ABSTRACT

Customer Relationship Management or in short, CRM, is a discipline as well as a set of discrete software and technologies which focuses on automating and improving the business processes associated with managing customer relationships in the areas of sales, marketing, customer services and support. Effective CRM software is deployed by organizations or suppliers that want to attract new customers and maintain current customers. More and more organizations have opted to spend their investment in CRM software as they believe it will generate more profits to the company. However, through findings and studies, it is found that more than 50% of the CRM implementation failed. This was due to five reasons such as system with no-ease-of-use, no understanding from business point-of-view, wrong focus, no user involvement and user resistance. This study attempts to clarify the issues that contribute to the failure of CRM and suggest how a Human Computer Interaction (HCI) approach will be able to address these issues. This study proposes to simplify the design of the system, the use of User Centered Design (UCD) approach, use of Hierarchical Task Analysis (HTA), use of metaphor and identify user’s behavior and characteristic in order to come out with an effective CRM system that can be implemented successfully. This study also had been carried out in a case study company that fits the criteria of supplier. The development and implementation of the system manage to address the current CRM issues and allow users to carry out their task efficiently and effectively thus allows the organization to satisfy their customers.
CHAPTER 1

INTRODUCTION

1.0 BACKGROUND

Customers nowadays have variety of choices and they are also becoming more knowledgeable and demanding. The power now had been shifted to the customer. With this current scenario, the company has realized that they need to treat their customer with care. Companies now have to figure out different ways to manage their customer effectively so that besides of acquire new customer, they will be able to retain their existing customer too. The emergence of competition has changed many aspects of existing business and generated companies with new business models, business opportunities and process. The existing companies are being challenged to rethink the most basic business relationship - the one between an organization and its customers. To overcome this challenge, many organizations are considering adopting the concept of Customer Relationship Management (CRM).

CRM is a discipline as well as a set of discrete software and technologies which focuses on automating and improving the business processes associated with managing customer relationships in the areas of sales, marketing, customer services and support (Customer Relationship Management (CRM) – Beyond the “Buzz”,2003). It is an overall business strategy that enables companies to manage the customer relationship effectively. It is a strategy used to learn more about customers’ needs and behaviors in order to develop stronger relationships with them. There are many technological components to
CRM. The more useful way to think about CRM is as a process that will help bring together lots of pieces of information about customers, sales, marketing effectiveness, responsiveness and market trends.

Due to its promising benefits, a lot of companies had been implemented the CRM system in their company. However, research and study showed that the benefits can only be gained for those that can get it right. According to Boardman, R. (2004), in the small and mid-size (SME) space, 90% plus of implementations fails to generate any significant pay-back on the initial investment made. Research and studies also showed that the rate of CRM failure is very high. According to Hershey, L. (n.d), between 30% and 75% of CRM implementation fails to produce the expected results and Return of Investment (ROI). Dozens of articles had been published, explaining the many reasons why CRM implementations failed. Based on the research conducted found that, the CRM failure was because the system is not user-friendly, user resistance and no user involvement. Thus by using the Human Computer Interaction (HCI) approach, will be able to address the current CRM issues.

This research projects looks into the issues that haunted CRM failures currently. It concludes with existing research that HCI approach is needed in order to make the implementation of CRM successful. It focuses on the sales forecasting process, which is one of the critical processes in CRM. It also proposes the use of User Centered Design (UCD) to include users as early as possible in the system development in order to avoid the user resistance in the CRM implementation. HCI approach is adopted to design an
interactive system in order to design an effective and efficient system to help user to be highly productive in their work. Besides that, it also proposes to simplify the design of the system as simple interface increases the intuitiveness, efficiency, and comfort level with a system, thus translates into system acceptance and use. Besides that, the Hierarchical Task Analysis (HTA) is also proposes in order to breakdown the task performed by the users in the forecasting process. Users in the proposed system are identified based on their behavior and characteristic. Other features have also been proposed for this system; metaphor, visualization and case-based reasoning (CRR). Based on all these features, the propose CRM software is developed.

For this research project, a case study is conducted on Continental Sime Tyre Sdn. Bhd (known as CST). It acts as supplier in this context. It is a Malaysian joint venture company between Continental AG of Germany, the forth largest tyre manufacturer worldwide and Sime Darby Berhad, Malaysia leading multinational and one of South East Asia’s largest conglomerates. The company is chosen as a sample as it wanted to adopt the CRM, fits the criteria as a supplier and also faces the CRM issues discussed in the above discussion.
1.1 RESEARCH MOTIVATION

The motivation of this research is initiated from the existing CRM system which can be widely found in the market but instead, are lack of high usability from user point of view. The failure rate of the CRM implementation had been a topic of discussion since the past years. It was failure attribute to end users who resisted of change, system with no high usability. Despite of its popularity in the market and promising benefits of generating more profits to the company, maintaining the relationship with customer effectively and efficient, and many more of interesting advantages, many companies have failed to implement it successfully and help them to gain the profits. The current CRM implementation did focus more on technology, and thus neglected the importance of usability and Human Computer Interaction (HCI) approach in the system. Thus these issues that contribute to the CRM failure were being identified and studied, and these factors will be incorporated inside a self developed CRM system which is simple but instead should be able to address the current issues that had been haunted the CRM implementation failures.
1.2 RESEARCH OBJECTIVES

There are two main objectives of this research project that must be fulfilled:

1. To develop a usable Customer Relationship Management (CRM) system approach using User Centered Design (UCD) in order to increase the usability.

2. To incorporate Case Base Reasoning (CBR) in developing a Customer Relationship Management (CRM) system in decision making process.
1.3 PROJECT SCOPE

The research project will cover the forecasting process in CRM from the supplier perspective. It will cover the process of forecasting consolidation made by Marketing in order to come out with the moderated list for the factory to produce the products demanded by the customers. All these will be using the User Centered Design (UCD) approach and other HCI approach and techniques which will be explained in details in Chapter 3 – Critical Analysis.

1.4 TARGET AUDIENCE

The target audience of the research system consists of 2 main parties, which are:

1. Marketing planner

These are the people who consolidate and moderate the orders and forecast based on demands, promotions, offers and trends. They are involved in forecast moderation module.

2. Production planner

The planner is the main key user whereby he will moderate the moderation list based on factory capacity as well as utilization considerations in the factory acceptance module. He will then come out with the factory plan on schedule to produce the output in order to meet the delivery date that had been agreed.
1.5 RESEARCH METHODOLOGY

There are six approaches are implemented in this research in order to develop guidelines for the proposed CRM system. The research methodology flow used in the research is as the following steps:

1. Literature review

While conducting the literature review, few CRM systems available in the market had been reviewed in order to know what are the strength and weakness of the systems, from Human Computer Interaction (HCI) point of view. Besides that, the current CRM issue that had been faced in the CRM implementations had been indentified so that all these issues will be able to be address in the propose CRM system. All the information values are found using the online search through the internet.

2. Data gathering

Since data gathering need to be performed to collect sufficient data in order to produce sets of complete requirement, thus few data gathering technique are identified. These techniques are combined and used so that the data gathering technique can be varied.

3. Capturing system requirement

Since use case had been proven as effective mechanism to capture the requirement, thus it had been use to capture the system requirement in this research project. For each module of the system, the use cases will be converted into a list of requirements in order for the modules to be implemented successfully.
4. Analysis and design

In this research project, the conceptual design gives attention to the system functions whereas the technical design expresses the form the system will take including the hardware and software aspects.

5. System testing

Different type of testing will be performed in order to make sure that it meets the user’s requirements.

6. System implementation

The system implementation is to make sure that the new CRM system is available for the use of users. The deploying of the system includes executing all steps necessary to educate users on the new system including user training and providing user manual for guidelines.

1.6 PROJECT SCHEDULE

Project planning is one of the factors that determine the success of the project. The schedule development is a crucial factor as it creates a realistic project schedule that provides a basis for monitoring project progress for the time dimension of the project. The schedule had been planned properly using the Gantt chart (refer to Appendix C – Gantt Chart). Gantt chart provides a standard format for displaying project schedule information by listing the project activities and their corresponding start and finish date in calendar format.
1.7 ORGANIZATION OF THE REPORT

This project is organized into seven related chapters.

Chapter 1 presents the introduction of this project. It starts by presenting the project definition, project objective, project scope, the target audience, research methodology, project schedule and organization of the report.

Chapter 2 provides literature review about customer relationship management (CRM), understanding user, user characteristic and discussion about the case base reasoning (CBR) technique and human computer interaction (HCI) approach inside the system.

In Chapter 3, the current CRM issues are highlighted. This chapter discusses how each of the CRM issues be addressed using HCI approach with additional features and concepts that the system should incorporate.

Chapter 4 discusses research methodology used to develop the system. It consists of the information strategy used in CRM, process model of system development and the technique used to capture user requirement.

Chapter 5 discusses the requirement analysis for the system. Each component of the system are identified and examined and treated as modules. This chapter discusses feature of the system which consists of the functional, non-functional requirement, usability requirement, system design and system model using the UML, database design and the graphical user interface of the system.
Chapter 6 discusses the design phase of the new CRM system using HCI approach, that are based on the critical analysis and requirements elicited in Chapter 3 and Chapter 5 respectively. The design phase consists of the design activities that point up the structure, method and system specification that satisfy the functional requirements developed in the requirement phase.

Chapter 7 discusses the system environment employed during the development and implementation of the CRM system using HCI approach. Then, it presents an implementation level view of the tool's operation. The explanations on codes are discussed to show how the functionality and features were implemented in the system. Besides that, the verification of each requirement and system design follows after each of the module interfaces is shown.

Chapter 8 discusses system testing in the development cycle of the system. It consists of unit testing, integrate testing and user testing.

Chapter 9 presents a conclusion to the work and provides future system enhancement for this research.
CHAPTER 2
LITERATURE REVIEW

2.0 INTRODUCTION

The steps in the evolution of management information systems are results of a large number of contributions which have brought progressive refinement of methodological approach to system management and the achievement of popular conceptual models such as Management Information System, Decision Support System and Executive Information System (Iivari, 1992). These innovations thus lead to modern Customer Relationship Management (CRM) system.

Table 2.1: Decision systems: Evolutionary stages and company process (Rajola, F., 2003)

<table>
<thead>
<tr>
<th>Evolutionary stages</th>
<th>IS</th>
<th>MI</th>
<th>DSS</th>
<th>EIS</th>
<th>BI</th>
<th>CRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex-post observation</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Management decisions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Low</td>
<td></td>
</tr>
<tr>
<td>Coordination</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Medium</td>
<td></td>
</tr>
<tr>
<td>Relations with the environment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>High</td>
</tr>
</tbody>
</table>

Company process
CRM is one of those amazing concepts that swept the business world with the promise of forever changing the way businesses small and large interacted with their customer bases. It’s started off in 1980’s with the emergence of database marketing that contributed to the setting up of customer service group whom will speak individually to all the company’s customer. The result, more often than not, was failure for several of reasons, including lack of organizational discipline, key staff departures prior to system completion and most importantly, a belief that computer systems were the answer to CRM woes. CRM failure was the rule rather than the exception for many early adopters, but lessons were learned along the way, including the reality that solid and uniformly embraced sales and business processes drive the relationship and sales effort. Certainly, technology can be critical to the successful support of good processes, but it can’t be the driver. (Harnish, M. & Casari, T., 2004).

Then, in 1990’s, most of the company began to improve their CRM by making it more of two way communications which is instead of only gathering the information for their own used, they are also start to reward back the customer by improving the customer service and also included some incentives, gift and others for the customer loyalty. By doing this, the CRM can be used as a way to increase sales passively, as well as through active improvement of customer service.

Nowadays, as more advanced solutions that were customizable across industries, CRM became a way to continuously update the understanding of customer need and behavior. The CRM software becomes more configurable, customizable, mobile, and
modular and process oriented. Advances in Internet utilization and security has brought new dimension in mobile CRM which is the hosted CRM solution. This solution is available through internet and maintained by the third party parties. The hosted CRM systems eliminate the need for costly hardware and software investments, system maintenance and upgrades and trained personnel staff. Today, CRM is still utilized most frequently by companies that rely heavily on two distinct features: customer service or technology. (Roberts, L.P.,2005)

In the late 80’s, the system interface are complex and driven by command line entries. Business software was developed by experts so the usability is not a primary concern. With the success of Microsoft Windows in the early 90’s has created new opportunities in the software design. By the late 90’s, usability became the differentiator (User-Centered Design: Making your Customers Work for You, 2007). Besides of depending on the developer intuition, the software vendors had started to proactively visiting user’s site to learn how the users are actually performed their tasks. End user adoption and ease-of-use contribute to greater success and lower costs of CRM rollout (SAP’s Next Evolution of CRM Delivers Customer-Inspired Innovation, 2005). More and more CRM vendors began to realize their importance and include them inside the software. By making it easier for users to extract information and perform business tasks, the latest evolution of the CRM user interface will help reduce task handling times, training costs and number of clicks.
According to Andersen, K. (2006), the importance of usability in CRM system is undeniable. It is important because it drives the user adoption. No matter how well the CRM system is designed, but if the customers did not use what is built-in inside the system, then the return on investment will fail to materialize. Thus, the user adoption rate should be considered as key business metric.

According to CRM: From evolution to revolution (2001), one of the CRM critical success factors that must be evaluate is to involve users as early as possible. Users knows their needs best and can help define the new system, as they are the one who are aware of the old system. And it is important to make them participate in the process and own the solution. The Smart Organization’s Guide to Implementing Change (2005) conclude that user centered design should span the entire CRM effort.
2.1 CUSTOMER RELATIONSHIP MANAGEMENT

Customer Relationship Management (CRM) is a method that integrates the concept of Knowledge Management, Data Mining and Data Warehousing in order to support the organization’s decision making process to retain long term and profitable relationship with customers (Cunningham, Yeol & Chen, 2004). CRM supports the organization’s decision-making process to retain long term and profitable relationships with its customers. Research by Winer (2001) suggest that 5% increase in retention can be as much as 95% boost in profit and repeat customer generate over twice as much as gross income as new customer (as cited by Brohman, Watson, Piccoli & Parasuraman, 2003, p. 47). Besides that, Massey.P. and Anne P. found that acquiring new customers could cost five times more than it cost to retain the existing customers (as cited by Cunningham, 2004, p. 14). Thus it is important for an organization to understand their customer’s current behavior, preferences and future needs. That is one of the reasons most company opts for Customer Relationship Management (CRM) as it is able to help the company build long lasting relationship with their customers.

Nowadays, organization/suppliers have realized that instead of treating all the customers equally, it is more effective to invest in customers that are valuable or potentially valuable, while limiting their investments in non-valuable customers. Suppliers must then respond to customer initiatives and comply within time frames that are established by the customers. Customers are important and their need must be well taken care off. Suppliers, in most cases deal with number of different customers, and therefore they meet a number of potentially conflicting demands and needs. Although
customers are pressuring suppliers to provide the best, for example to communicate electronically and to automate all the processes, but the compliance should not be the supplier’s primary concern. Even though the customers want to be served according to their individual and unique needs and demands, the companies need to develop and manage the relationships with their customers so that the relationships are long-term and profitable. The main focus should be doing what’s best for supplier’s business. Therefore, companies are tuning to Customer Relationship Management (CRM) techniques and CRM-supported technologies.

Suppliers need to understand the benefits that can be gained by transacting the business and automating the process and eliminating the manual processes. Besides that they also want their transactions to be secured and reliable, beside of being careful to avoid the investment that will impede their adaptability and flexibility. Furthermore, they want to be assured that they can migrate at their own pace. All these can be accomplished by investing in solutions that will not tie them to specific technology that will consume capital and other resources as they find it necessary to upgrade and meet future demands.

According to Brohman’s et al. (2003), CRM success can be defined in three dimensions which are:

a. Increase profit
b. Improved customer satisfaction
c. Enhanced customer loyalty
Different orientation has different approaches toward examining the customer data (Brohman et al., 2003):

a. Profit-centric orientation

Profit-centric orientation to CRM concentrates on capturing and analyzing historic transaction and preference data for existing customer. Data is used to identify the most profitable customer who is expected to continue to contribute to the company’s future. As for the unprofitable customer, no investment is made on these groups.

b. Customer-understanding orientation

Customer-understanding orientation to CRM strives to understand the needs and preference of current and potential customers and uses this information to better service them. By doing this, the company benefits not only from the existing customer who already provides the revenues but also supports the customer who has the potential in the future.

c. Customer-relationship orientation

Customer-relationship orientation focuses on managing individualized relationships with the customers. The company and the customer mutually benefit from the close relationship which develops from the successful practice of the CRM.
Table 2.2: Definition of information processing strategy (Brohman, M.K., Watson, R.T., Piccoli G., & Parasuraman, T., 2003)

<table>
<thead>
<tr>
<th>Information Processing Strategy</th>
<th>CRM Orientation</th>
<th>Desired Outcome</th>
<th>Primary Type of Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction</td>
<td>Non-CRM, transaction-based only</td>
<td>Efficiently handle transaction-based customer inquiries</td>
<td>Firm-specific current behavior</td>
</tr>
<tr>
<td>Data</td>
<td>Profit</td>
<td>Increase profit per customer</td>
<td>Firm-specific past behavior and preferences</td>
</tr>
<tr>
<td>Inference</td>
<td>Customer understanding</td>
<td>Capitalize on cross-sell and up-sell opportunities</td>
<td>Multiple-firm past behavior and current preferences</td>
</tr>
<tr>
<td>Advice</td>
<td>Customer relationship</td>
<td>Develop long-term customer relationships</td>
<td>Future needs</td>
</tr>
</tbody>
</table>

Based on the research, customer-relationship orientation best suit the company in the case study due to the fact that the company (in the case study) would be able to treat the customer individually and uniquely rather than calculating profit from a discrete transaction.
Figure 2.1: CRM Information processing strategies and supporting customer views

2.2 NEW APPROACH OF CRM

According to Gray, P. and Byun, J. (2001), CRM consists of four major modules which are:

a. Sales Force Automation (SFA)

b. Customer Service (CS)

c. Sales and Marketing Management (SMM)

d. Contact and Activity Management

Regardless of modules implemented, the primary goal of CRM is to develop customer loyalty and generate more business that is more profitable. In order to achieve these goals, the system must be implemented successfully. If the users do not know how to best use the application or get what they want to perform the tasks, then these will lead to user frustration and do not allow them to perform their task efficiently and effectively and thus make the goals of the CRM unreachable. Currently, many CRM vendors had seemed to get feedback from their users and customers through surveys, the Internet and other methods. But all these are not done through on-site interaction. Even though the users are being consulted in these processes, this is not user-centered approach (UCD). UCD refers specifically to the process of observing the user in action and applying their input to evolve designs. The involvements of the users are included right in the early stage of the system development until the system implementation. The outcome of this approach resulted in greater user satisfaction, higher percentage of repeat business and higher revenue per customer and thus helps to implement the CRM successfully.
Besides that, creating the user experience for the business application is the key to the success of the CRM implementation. According to Jacko, J.A and Sears, A. (2003), Human Computer Interaction (HCI) is critical success of CRM. The human factor plays an important role in the design of the system. Another technology that has proven to be important is the appearance of visual interface. According to Business Systems Group (2006), the visual interface can account for more than 40% of the development effort for an application. Nowadays, lots of CRM vendors have come-out will a very high-tech features and interfaces which may look very impressive but later lead to users dissatisfaction since these features are hardly use by them. Products that are easy to use by users will be able to achieve differentiation in competitive market. A simple visual interface make the user more effective and efficient in performing their task as they will not spend more time struggling with a poorly designed user interface and poorly throughout the functionality in the system.

Like any other things, CRM systems also have few shortcomings that need to be addressed. Among of its shortcomings is the system implementation that does not lived up to the expectation due to the large part to project scope (Miller, F, 2006). Besides that, most of its system implementation failed due to how the system was implemented. In most of the system implementation, only the management made the decision without involving user in the selection and implementation process. This thus led to the no sense of ownership of the changes being attempted to their work processes (Bennokom, F. V. and Blaidsdel, M., 2007). It has lots of areas of improvement. Product development begins with a vision of a product, which includes a vision of the users for that product.
Therefore, it is critical to accurately understand end users' needs. Traditionally, software development has been technology-centered and feature-drive. Today, when designing the system, the central concern is not hardware performance but user performance and satisfaction. The system should be efficient, easy to use in order to improve the user’s productivity and overall experience. The selection of good models when developing the CRM product that involves the users throughout the software development and implementation would be able to address the problems faced currently and thus able to avoid the CRM failure.
2.3 CASE STUDY: CONTINENTAL SIME TYRE SDN. BHD.

A case study had been conducted in a multi national company (MNC) in Malaysia, called, Continental Sime Tyre Sdn. Bhd. or known as CST. It is a Malaysian joint venture company between Continental AG of Germany (the forth largest tyre manufacturer worldwide) and Sime Darby Berhad (Malaysia leading multinational and one of South East Asia’s largest conglomerates). CST is the holding company with its tyre manufacturing business activity carried out by two tyre manufacturing companies, Continental Sime Tyre PJ. Sdn. Bhd (CSTPJ) with its factory located at Petaling Jaya, and Continental Sime Tyre AS Sdn. Bhd (CSTAS) located at Alor Setar.

The group basically involved in the manufacturing and marketing of wide range of tyres for motorcycle, passenger car, four wheel drive, truck, bus, forklift, earthmover and agricultural, industrial and military vehicles It has the largest tyre dealer network totaling 1000 tyre outlets providing a wide range of car care and tyre services to motorists throughout the country. Its tyre brands include Continental, Dunlop, Barum, Uniroyal, Simex and Sime Tyres. It now ranks Malaysia’s No 1 producer of passenger tyres for the original equipment and replacement business and occupies first place in the replacement business for commercial vehicles tyres.

In order to value the relationship with the dealers, CST normally recognizes the achievement made by the dealers. Last year, in Jun 2005, 30 dealers had been awarded the recognition to tour the Europe for nine days after achieved sales turnover required by the company. Besides the recognition award, the company had also conducted Network
Dealers Retail course whereby in this course, the participants, mainly dealers had been trained with new skills and knowledge to keep up with the changes in the market place as well as to raise the standard of service in al Dunlop Klinikar and Pusat Sime Tyre outlets. The company had mapped out a long term training and development strategy for the dealers. This is in line with the vision of Continental Sime Tyre to produce the best dealers and create the most conducive tyre service centre throughout Malaysia.
2.4 FORECASTING PROCESS

This research will be focusing on the forecasting process in CRM. Nearly every CRM implementation stresses the importance of forecasting. According to ASW Customer Delivery Schedule (2003), forecast can be handled in various ways.

i. **Net forecast**

In this type of forecast, the customer determines the dates and quantities of the goods. There will be no adjustment made to the quantities in the delivery schedules. The supplier will just deliver what the customer requests.

ii. **Gross forecast**

The gross forecast contains of min/max/actual inventory. The available quantity is calculated as the actual inventory plus the adjustment quantities.

The information from forecasting process can be fed to optimize the master production schedule or factory plan and thus will be able support materials resource planning (MRP) and production scheduling.

The forecasting module had been identified to be developed in the test sample company, CST. In today’s competitive market, the ability to develop an affective sales forecast manage to give impact to the company’s benefits. Forecast serve as a basis for planning expenditures from marketing and product development to new capital equipment.
In CST, the forecast sale estimate (FSE) is done by tyre size is submitted by each sales office by mid month on every month. The forecast is done based on the market group: Replacement market, export market, original equipment (OE) and contact market. Consolidation of the sales estimates need to be done in order to work out the quantity of tyres need to be manufactured.

A decision making will takes place when the marketing plan moderates the forecast quantity to manufacture based on the few contributors as the following;

i. **Stock**

   Based on the availability of the stock, a decision need to be made in order to determine which particular tyre size requires production. If the quantity of stock is enough to meet the quantity of tyres demanded by the customer, therefore there is no need to produce one.

ii. **Past month sales**

   Good past month sales shows that there was a good sale for the previous month and therefore a decision can be make to increase the quantity of tyres that need to be produce.

iii. **Demand**

   The state of demands depends on many external factors which need to be predicted. High demands of tyres is occurred when the festivals season are around
the corner (when people start changing car tyres for long distance traveling), company starts doing promotions, the price reduction in tyres and so on.

iv. Factory capacity

Factory capacity is the state of the production which depends on few factors such as machine capacity, the availability of manpower, tooling, raw materials and the capacity utilization days (CUD). CUD is the number of days that the plant has which is based on factory efficiency after consideration of public holidays (personal communication, November 10, 2005).

v. Popularity

Popularity refers to the tyre size popularity which means that which particular size is famous among the customers. The popular size is given high priority to be produce because normally these tyres are in great demands and the quantity of tyres to be produced is increased as compared to the original forecast quantity.

Upon the completion of moderation process, a meeting is done between marketing and production to agree on the quantity of the tyre to be produced that had been set in the moderation process. An agreement need to made to determine amount of finalized quantity that need to be done or any tyre sizes that are going to be reduced or dropped. The production planner will then confirm and finalize all the sizes to be manufacture based on the factory capacity and constraints. The weekly production schedule is produced based on the agreed program in the meeting. The production team have to
manufacture the tyres according to the plan and the aim is to deliver the finished goods to the store. If any unforeseen constraints and problem arise on the shop floor, therefore there is a need to revise the factory plan.
2.5 HIERARCHICAL TASK ANALYSIS (HTA)

Hierarchical Task Analysis (HTA) will be used in this research to describe the process of producing the sales forecast in CRM. HTA is a method decomposes high-level task into the subtask and actions based on graphical structure chart notation (Shepherd, 1989). It involves an iterative process of identifying tasks, categorizing them, breaking them down into subtask and checking their accuracy. It is a very useful for HCI work. It is done by performing the observation on users; how they work and steps required in completing the tasks. Asimakopoulos, S., Fildes, R. and Dix, A. (2005) had used HTA to describe the normative process of producing the sales forecast. The following shows the example of the HTA for borrowing a book from the library:

0 In order to borrow a book from the library
   1 go to library
   2 find the required book
      2.1 access the library catalog
      2.2 access the search screen
      2.3 enter search criteria
      2.4 identify required book
      2.5 note location
   3 go to collect shelf and retrieve book
   4 take book to checkout counter
Figure 2.2: Example of HTA diagram (Rogers, Y., Sharp, H. & Preece, J., 2002)
2.6 HUMAN COMPUTER INTERACTION (HCI)

Study conducted by Myers, B. A. (1998) showed that research in Human Computer Interaction (HCI) has been successfully and fundamentally change the computing worlds today. The ubiquitous graphical interfaced by Microsoft Windows 95 was based on Macintosh that is based on work at Xerox PARC. Even the growth of World Wide Web is a direct result from HCI research. The now ubiquitous direct manipulation interface where visible objects on the screen are directly manipulated with a pointing device was first demonstrated by Ivan Sutherland in sketchpad in 1963. Multiple tiled Windows were demonstrated in Engelbart’s NLS in 1968. The idea of hypertext is credited to Vannevar Bush’s famous MEMEX idea from 1945. Even the User Interface Management System (UIMS) was first created at Imperial College London by William Newman (1966-1967). Therefore we can conclude that all of the most important innovation in HCI have benefited from research done by the HCI experts in the early days.

Early CRM system involved a large software packages that often requires years to implement. At this time, the mobility was not adequately addressed as well. At this point in time, the CRM software was not configurable, customizable, mobile, modular and process oriented. Thus, its make the CRM application not user-friendly at all as it makes the live of users more difficult when using the automated system as compared to when they are performing the tasks manually.
HCI is a discipline concerned with design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them (ACM SIGCHI, 1992, p. 6). According to Butler, Jacob and John (1999), design in HCI is more complex than in many other fields of engineering. It is an interdisciplinary area that influences diverse and different kinds of areas: computer science, software engineering, human factors and psychology. Even though they are coming from different disciplines having different focus and methodology, but all are concerned with designing the system to match user’s goal. By having an interactive software or product which is easy, effective and enjoyable to use, will then make then make interaction more between the computer and human beings.
2.6.1 User Centered Design (UCD)

User-centered design or UCD, is a comprehensive software development methodology driven by clearly specified, task-oriented business objectives and recognition of user needs, limitations and preferences (User-Centered Design Methodology, n.d). There is need for a user-centered approach in software development that focus on real users and their goals to drive behind the development of the product. As a consequence, a well-designed system should make the most human skill and judgment, should be directly relevant to the work in hand, and should support rather than constraint the user (Roger, Sharp & Preece, 2002). Nowadays, more and more company are realized the importance of the UCD and come out with the own UCD process.

According to Rubin (1984), in UCD, all development proceeds with the user as the center of focus. Product development which is based on user’s needs cannot succeed. By talking to users we would be able to uncover specific requirements for existing software, ideas for new products and even inspire revolutionary innovation. Rubin depicts the UCD process as the following:

a. Users are in the center of a double circle

b. The inner ring contains the Context, Objectives, Environment and Goals

c. The outer ring contains the Task details, Task Content, Task Organization and Task Flow
There are different degrees of user’s involvement in the product development, starting from requirement study, design, evaluation studies until testing of the product. According to Roger et.al (2002), these different degrees of user involvement may be implemented in order to manage expectation and to create a feeling of ownership. Users may be opted for full time basis or part-time basis and it may be for the duration of the project of for a limited time only. There are advantage and disadvantage to each situation. If user is co-opted for the whole project, their input will be consistent and they will become very familiar with the system its rationale. However, if the project takes many years, then they might lose touch with the user group that makes their input less valuable. If a user is co-opted part time for the whole project, they will offer consistent input to development while remaining in touch with other users. It depends on the situation. Instead of involving users directly in the development, they may only be kept informed through regular channels of communication such as emails and newsletter.
The stages in UCD are carried out in an iterative activity, with the cycle being repeated until the project’s usability objectives have been gained. The UCD approach demands design iterations, so far better solutions emerge in less time. The overall screen-based customer experience grows by iterations of creative problem solving and structured customer feedback. The test, prototype, test and redesign process is highly iterative and may involved several cycles.
2.6.2 User Centered Design Benefits

The UCD has numerous benefits for business. Among of them are:

a. Software development using UCD methodologies is more efficient. Usability is easier and less costly to incorporate earlier in the product life cycle.

b. Lower documentation and support costs. Product and accompanying documentation created with UCD reduced the number, length and severity of customer support calls.

c. Improve customer satisfaction. Products which create a positive user experience are easier to use and inherently more satisfying because their products enable users to achieve their goals more easily and thus, increase satisfaction.

d. Provide competitive advantage. As technologies and products mature, often offering comparable features, superior user experience and usability distinguish the UCD products from others with similar functionality.

e. Reduce development costs. Intensive engagement with users, learning what they want and how they work will speed design decision making, prevents false starts and minimize costly redesign.

The use of UCD in developing CRM system will be able to open up a two way dialogue between a company and its customers. In UCD, it would be able to unstated needs and provide the solution to address these needs. This thus will lead to the increase of user’s satisfaction across the customer’s base.
2.6.3 ISO 13407: Human-Centered Design Process

There is an international standard that is the basis for many UCD methodologies. ISO 13407 standards provide a framework for user centered development activities that can be adapted to numerous development environments: from a straight waterfall type of development process to an iterative type of environment (refer to figure 2.2). Human centered design processes for interactive systems, ISO 13407 (1999) states the following:

“Human-centered design is an approach to interactive system development that focuses specifically on making systems usable. It is a multi-disciplinary activity”

![ISO 13407 Model Overview](EMMUS, 1999)

The model comprises of five stages, four of which are implicitly joined in a loop. Although the process outlined above looks iterative, it does not necessarily need to be followed as it may be converted to a waterfall life-cycle model if required by simply
going through once only or a V-type lifecycle development. In this model, once the need to use the UCD process had been identified, four activities form the main cycle of work:

i. Specify the context of use - Identify the people who will use the product, what they will use it for, and under what conditions they will use it.

ii. Specify requirements - Identify any business requirements or user goals that must be met for the product to be successful.

iii. Create design solutions - This part of the process may be done in stages, building from a rough concept to a complete design.

iv. Evaluate designs - The most important part of this process is that evaluation - ideally through usability testing with actual users - is as integral as quality testing is to good software development.

The true benefit of this model emerges when it is used to guide an iterative development process.

According to Smith and Reinersten (as cited by Marcus, 2002), the first 10% of the design process, when the key system-design decision are made, can determine 90% of a product’s cost and performance, usability techniques help keep the products aligned with company goals.
2.6.4 User-Centered Activity

The UCD activities are broken down into four phases, which are analysis, design, implementation and deployment (What is User-Centered Design?, n.d).

i. Analysis phase
   a. Meet with key stakeholders to set vision
   b. Include usability tasks in the project plan
   c. Assemble a multidisciplinary team to ensure complete expertise
   d. Develop usability goals and objectives
   e. Conduct field studies
   f. Look at competitive products
   g. Create user profiles
   h. Develop a task analysis
   i. Document user scenarios
   j. Document user performance requirements

ii. Design phase
   a. Begin to brainstorm design concepts and metaphors
   b. Develop screen flow and navigation model
   c. Do walkthroughs of design concepts
   d. Begin design with paper and pencil
   e. Create low-fidelity prototypes
   f. Conduct usability testing on low-fidelity prototypes
   g. Create high-fidelity detailed design
h. Do usability testing again

i. Document standards and guidelines

j. Create a design specification

iii. Implementation phase

a. Do ongoing heuristic evaluations

b. Work closely with delivery team as design is implemented

c. Conduct usability testing as soon as possible

iv. Deployment phase

a. Use surveys to get user feedback

b. Conduct field studies to get info about actual use

c. Check objectives using usability testing

The process ends and the product can be deployed once the requirements are met.
2.7 HCI DESIGN PRINCIPLES

Don Norman (1998) in his bestseller *The Design of Everyday Things* had described the most common design principles. They are:

i. Visibility – the more visible the functions are, the more likely users will be able to know what action to be done next.

ii. Feedback – sending out the information on what actions has been done in order to allow the users to continue with the next activity.

iii. Constraints – it is type of user interaction that restricts users from performing certain tasks.

iv. Mapping – refers to the relationship between controls and their effects in everyday life.

v. Consistency – refers to designing interfaces to have similar operations and use similar elements to achieve tasks.

vi. Affordance – used to refer to attribute of object that allows people to know how to use it (Roger et.al, 2002).

Good design will be able to allow users to be highly productive in their work. Users is a person with a whole set of expectation and predilections. In order to
understand what are the thing that user need, therefore we need to understand the
characteristic of the user and their capabilities, what they are trying to achieve, how they
achieve it currently and whether they would achieve their goals effectively if they were
supported differently.
2.7.1 Type of Users

User can be categorized into three groups which are:

a. Novice user - Systems to be used by novices require more feedback and more opportunities for closure so that the novice feels that progress towards the goal is being made and not left for long periods of time wondering if what had been done so far is correct. They will require step-by-step instructions, probably with prompting and a constrained interaction backed up with clear information.

b. Knowledgeable intermittent - These groups of user are able to maintain semantic knowledge or both the task they want to perform and know about the computer concepts involved.

c. Expert/Frequent users - Expert users are well versed in both semantic and syntactic aspects of the computer systems and their response time are rapid. They will require a flexible interaction with more wide ranging powers of control.
2.7.2 User’s Physical Capabilities

There are many dimensions along which a user’s capabilities may vary and that they will have an impact on the product’s design. For example, a person’s physical characteristic may affect the design: size of hands may affect the size and positioning of input buttons. According to Busby and Edwards (1997) (as cited by Keates, Langdon Clarkson & Robinson, 2002), users with a number of different physical impairment conditions have the same desire to use computers as able-bodied people but cannot cope with most current computer access systems. Thus it is important to identify the difference in interactions of differing physical capability.

a. The vision

For most human beings, the most important sense is a vision, by using vision; they have the ability to see. If we ask somebody what the unfamiliar object is like, he/she might probably describe in terms of its visual appearance first and other characteristics such as its taste, smell a little later. This shows that appearance of object is important to human beings. When we talk about something pleasant, we mostly mean its nice look. Thus sight is the most important senses for the development of human-computer system at this present time.

b. Colour

Any normal human being with normal colour vision is able to distinguish over 7 million different shades of colour. On the other hand, only eight to ten different colours can be identified accurately. However it is unwise to produce a computer system that
used all these colours at once. It is advisable to use five to seven colours with appropriate contrast (Myers, 1999).
2.7.3 User’s Mental Capabilities

Human memory consists of series of connected system. Memory involves recalling various kinds of knowledge that allow us to act appropriately. It enables us to do many things such as remember someone’s name or recall when we last met them. Without memory, we would not be able to function. It is not possible to remember everything that we see, hear, taste or touch, as our brains would be completely overloaded. A filtering process is used to decide which information gets further processed and memorized. Human memory consist three types of memories which are:

i. Long term memory (LTM) - Long term memory is where memories are stored.

ii. Working memory - Working memories is effectively where conscious thought processes operate.

iii. Sensory memory - Sensory memories are the area of memory which copes with input from the sensors.

Figure 2.5: Human information processing – a model of memory

a. How the memory work

i. The digit span

Human societies tend to classified as having good or bad memories. A technique called digit span had been introduce a London school teacher, J. Jacobs, in 1887. The digit span is determined as follows (refer figure 2.4). The subject is presented with a
series of digits and was asked to repeat them back in the same order. The number of digit is gradually increased until finally the individual fail to recall the list accurately.

![Figure 2.6: The digit span](image)

ii.  **Chunking**

George Miller’s (1956) theory that 7 plus or minus 2 chunk information can be held in short-term memory at any time. More items can be remembered if the items can be chunked. This means that pieces of information are group together so that they form one item to be remembered. This means it would be advisable not trying to present more than 7 plus or minus 2 to the users for them to recall for example, password of more than 5 – 9 characters would be difficult for most people to be remembered unless chunking is allowed. This chunk is not always obvious so that what is easy for one person is difficult
to another person. Memory capacity varies from person to person and even within individual there are variations. Some people have good memories for faces, some for numbers and so on.

Chunking means the grouping of information into sections. It will make sense to the individual and can be seen as entity by that individual. For example, the following numbers might be difficult for us to remember if we treated them as one entity.

154638956792

However, if we chunk the list into four groups of four, then we will find it much easier to remember.

638 956 792

By chunking words into sentence fragments, we are able to process an even greater amount of information in Short Term Memory (STM). Similarly, we will find it difficult to remember the following letters:

G G N N T I H I A U

But we will find it much easier to recall the following:

H U N G A T I N G

The reason why HUNGATING is easier to recall than GGNNTIHIUAU is that the letters of the HUNGATING can be divided into two section which is HUN and GATING.

iii. Closure

Human beings like to know when the task had been accomplish since it means that they are free to move on to the next stage and forget about the task that they had just
done. It is difficult to concentrate fully on another task when we are desperately tried to remember something that is not easy to remember. Therefore, it is important to build closure into the computer system because it acts as a means of allowing processes to be grouped and chunk in memories. The act of providing closure will also be a mean of giving feedback to the novice user and this will aid the learning process and reduce the burden on memory.
2.8 DIFFERENT TYPES OF CRM SYSTEM

There are several types of CRM applications from entry level contract management solutions to sales force automation, all the way to full blown package that interface with inventory system. There are so many off-the-shelf CRM packages available in the market. Some of them are large scale application that aimed for big companies while some are targeted for the small-medium businesses. Most of the CRM vendors stress too much on the technology without having users in mind. Among of CRM packages that had been reviewed are:

i. Salesforce.com

Salesforce.com is the leading on-demand CRM tool that provides the benefits of Web-based application and have good user interface. It’s functionally covers the Sales Force Automation (SFA), Marketing and, Service and Support.

The forecast module in the system gives organizations clear visibility into their sales pipelines. By having accurate and timely forecast thus will help the sales close more deal, bring higher profits and align the expenses with the revenue growth. The forecast methods are flexible in which they can define it based on monthly or quarterly basis. It can be based on opportunity-based, product-based or schedule-based products methods. The module allow allows user to have custom period and views in which it allows user to make their own forecast view and choose the view numbers (in million, percentage etc). It also allows the managerial overrides where it allows the opportunity owner and their managers to apply their own subjective assessment of each opportunity. The report data
can be exported to CSV format and displayed in Microsoft Excel using salesforce.com Office Edition that allows user to manipulate data on their own. Besides that, the system also has a good interface, in which in one of the modules, it utilizes the Google Map that allows user to map the location of the contacts.

![Screen snapshot from Salesforce.com](image)

**Figure 2.7: Screen snapshot from Salesforce.com**

Salesforce.com does not have order management and the sale process ends at opportunity. Besides that, it does not have service level agreement (SLA) for most customers, which means if salesforce.com goes down, then the employee cannot access the customer’s record and no refund will be paid. In term of reliability, the system does not guarantee the uptime. According to ‘Salesform.com Need to Focus on Fundamentals’ (2006), it already has two service outages with the one lasting nearly six hours recently. Furthermore, it also has performance slowdown in recent month.
ii. Siebel CRM On-Demand

Siebel launched the CRM On-Demand with collaboration with IBM. Their entry into hosted, monthly CRM solution niche hit the marketplace with gale force. The application is a full-suite service, providing sales, customer service and marketing components. Sales CRM functionality includes forecast, opportunity, account, contact, and quota management: integrated sales analytics; and off-line access for the mobile or traveling user. Its customer service functions include problem management, knowledge base and integrated customer service analytics. Its marketing functions include the campaign management, integrated marketing analytics and outbound email. It is based on secure, flexible, high-performing Oracle technology.

In forecast module, the system eliminates the time-consuming manual forecast and allows the forecasting to be done on weekly and monthly basis. It allows the central location for all the opportunity data. Besides that, the system has features, by using the homepage customization that allow users to customize their own page and enable users to rearrange elements such as lists, history, favorites, user interface themes and analysis. The process flow of the system is shown in sequence in which user is allowed to click on Next button to follow the process step-by-step.
The system does not have good interface. All the details are put in the same page and user need to use the mouse and page down button to scroll to the bottom of the page. Besides that, no confirmation messages were prompted when user performed certain important tasks, such as saving the transaction. The important tasks buttons are place next to another, for example Close button is place next to Save button. This can lead to dangerous error as user might want to click on Close button, but accidently click on Save button. And since no message is prompted for confirmation, then the record will be saved. On top of that, whenever user performed certain task, they are force to go to multiple different screens, which might be confusing to new users. Besides that, the interface is done based on task-based. The task-based interface should be design to increase the productivity especially to novice users by guiding them through the unfamiliar tasks.
iii. NetSuite CRM

The NetSuite web-based application is stored centrally in Oracle database by eliminating the redundant data entry by ensuring consistent and up-to-date customer information. Its functionality covers from lead to order process, including the forecast. The system guarantees the uptime of 99.5%. It also has the order management where it allows the order to be placed to complete the sale. The information in the system is grouped systematically in which there was put in different tabs based on their functionality. Besides that, the important-action button, such as Save and reset button are separated with less-important-action button.

Figure 2.9: Screen snapshot from NetSuite CRM

The system does not have a good feedback features in which no message is prompted for confirmation when user perform important task such as saving the transaction.
Case-Based Reasoning refers to both a cognitive and a computational model of reasoning by analogy to past cases. A basic premise in CBR is that many problems that decision makers encounter are not unique, but rather they are variations of a problem type. It is often more efficient to solve a problem by starting with the solution to a previous similar problem than it is to generate the entire solution again from first principles. In fact, experts have been observed to reason by analogy to prior cases.

In solving a current problem, a case-based reasoner (whether it is a human or a computational model) recalls a similar past case and its solution. The reasoner then adapts the successful solution of the recalled case to adjust for any differences between the current case and the recalled case. Finally, the CBR stores the solution to the current case along with feedback about the outcome so that it can be used in solving future problems.

Typically, a CBR system consists of a data base of past cases and their solutions, a set of indices for retrieving previous cases and storing new cases, a set of rules for measuring similarity, and rules for adapting recalled case solutions. A CBR system first gains an understanding of the problem. This is accomplished by collecting case attribute values that identify the problem type and that distinguish one problem type from another. The case attributes that identify the problem type are used as indices for case storage and retrieval. Indices and rules for measuring similarity focus attention on the important features of a problem, i.e., features that can be used to explain why case solutions differ.
Once the CBR understands the problem, it is reminded of previous similar cases. Solutions to recalled cases provide possible solutions to the new case. In interesting problem domains it is unlikely that an exact match will occur, therefore, the CBR must adapt its solution. Adaptation rules capture domain theory about the impact of attribute values on the solution. Learning takes place when new cases are solved and stored in the case base together with the outcome of the solution. Learning also occurs when failed solutions are attributed to specific case features and those features are then added as indices. Solutions are attributed to specific case features and those features are then added as indices.
2.10 CONCLUSION

Based on the literature review, we can conclude that the customer relationship management system is essential in today’s competitive advantage. By using a CRM system, a company will be able to differentiate themselves from others by providing the good quality of services and their relationship established with the customer. With the combination use of different approaches of User Centered Design (UCD) in Human Computer Interaction (HCI) and Case Based Reasoning (CBR) methodology will be able to come out with a good CRM system to provide an interactive system and ease-of-use feeling besides of simplify the decision making process.
CHAPTER 3

CRITICAL ANALYSIS

3.0 BACKGROUND

The literature review discussed in Chapter 2 shows that Customer Relationship Management (CRM) is really crucial in today’s competitive business. Many organizations have opted for CRM system in order to retain long term and profitable relationship with their customers. However a lot of studies found that, over 50% of the CRM projects failed. Thus, Human Computer Interaction (HCI) approach is believed to be the answer of the issues addressed. This chapter will clarify the current issue that leads to the failure of the CRM projects and explains how the HCI approach can be used in the system that will help to address the issue. A study will be conducted in order to find what caused the failure of the CRM projects and ways to address the issue. The other important concepts and features that are relevant to the development of the system will also be proposed in this chapter.
3.1 ISSUES WITH CRM SYSTEM

According to Doomed From The Start (Boardman.R, 2004), 70% of the CRM projects failed and in the small and mid-sized enterprise (SME) space, 90% plus of implementations fail to generate any significant pay-back on the initial investment. A Garther study found that approximately 55 percent of all CRM projects failed to meet software customer’s expectations. In a Bain & Company survey of 451 senior executives last year, CRM ranked in the bottom three categories among 25 popular tools evaluated for customer satisfaction (Mello, A. 2002). Based on research conducted by Hee, W. K. and Shan, L.P. (2006) found that the failure rate of CRM implementations is estimated to be greater than 65%. Lowering the failure rate and supporting the success of the system is the ultimate goals of researchers.

According to CRM Failure Rate Top-Of-Mind (n.d), Crossware director Ken Fairgray says:

Any organization looking to implement a CRM solution needs to face up to some pretty daunting statistics. We sit down with senior management and explain the failure rate of CRM projects, he says, which is typically around the 70% mark.

3.1.1 System with No Ease-of-use

Most of the CRM applications are designed with complicated tools and functionality which somehow looks impressive but when it comes to using, users will face a difficulty. Users have to spend more time for training as well as to master the complexity of the processes. The complicated user interface of the CRM software, Aegis
tracking system, was a contributing cause to the erroneous downing of an Iranian passenger plane, and the US Stark's inability to cope with Iraqi Exocet missiles was partly attributed to the human-computer interface (Neumann, 1991)

According to Law Firm and CRM System (2001), not many CRM systems had been implemented in law firm. The law firm has unique requirements that most CRM systems do not address and most of the system are not easy to use. In law firm, CRM tool can be used by the lawyers and secretaries for limited purposes, not enough for them to become proficient at complicated functions. And neither lawyers nor secretaries can afford weeks or even days of training. Thus the system designed must be exceptionally easy to use and requires minimal training. The current CRM system lacking such features gain little acceptance and provide very limited value.

3.1.2 No Understanding from Business Solution’s Point of View

Study conducted by Boardman, R. (2004) found that software providers fail to talk about their technology in the context of a business solution at all. They understand the features and functions, but have no capability to see things through the eyes of the customer and communicate how the technology will solve the customer’s problem. Thus, when the system was implemented, it had left the un-experience client to try out the system, and often fail to work with their business solution.

3.1.3 Wrong Focus

When CRM vendors focus on the benefits of the technology, they focus on the wrong thing. They looked for technology that look powerful but don’t pan out in a live
implementation. According to Problems in CRM (n.d), users term CRM as ‘technology that delays or stops service’. Some of them find the automated voice system or interactive voice response as a big headache and look for companies which have ease of use.

3.1.4 No User’s Involvement

According to Standish CHAOUS Study in 2002, lack of user involvement has been the second reason for project failure. Conversely, it has been the leading contributor to project success. Even when delivered on time and on budget, a project can fail if it doesn't meet user needs or expectations.

Table 3.1: Recipe for Success: CHAOS Ten

(Project Management: The Criteria for Success, 2001)

<table>
<thead>
<tr>
<th>Confidence Level</th>
<th>Success Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive support</td>
<td>18</td>
</tr>
<tr>
<td>User involvement</td>
<td>16</td>
</tr>
<tr>
<td>Experience project manager</td>
<td>14</td>
</tr>
<tr>
<td>Clear business objectives</td>
<td>12</td>
</tr>
<tr>
<td>Minimize scope</td>
<td>10</td>
</tr>
<tr>
<td>Standard software infrastructure</td>
<td>8</td>
</tr>
<tr>
<td>Firm basic requirement</td>
<td>6</td>
</tr>
<tr>
<td>Formal methodology</td>
<td>6</td>
</tr>
<tr>
<td>Reliable estimates</td>
<td>5</td>
</tr>
<tr>
<td>Other criteria</td>
<td>5</td>
</tr>
</tbody>
</table>

A lot of CRM developments only include managers and top level staffs during their analysis. These groups of people usually pull statistics from the system but don’t enter information from the start to end of a cycle. Kujala and Kauppinen (2004) had
described a case study in which designers selected user under the guidance of the manager. Later, it found that the needs of other users (such as junior, part time and temporary clerks) had been neglected. This resulted in a situation whereby these left out users did not see the good reason to use the resulting system and thus rejected the system.

3.1.5 User Resistance

According to research made by Lombardo (2003), some of the tools in CRM tend to have many features which are too complicated for the users. Management does not take the time to explain how these tools will be able to improve their lives and able to increase their productivity. Beal (2003) provides review on CRM implementation that one of the failures in the CRM implementation is user resistance to adoption. In Siebel System implementation in Parametric Technology Corp (PTC), they have adopted all the things that the industry analyst say make a successful implementation and yet still receive twenty percent of the users who makes noise and reluctant to use the system. Even though the numbers looks small, but if even one sales representative is not using the system, it will disrupt forecasting for the entire organization. According to research conducted by AMR Research Inc on 80 end-user companies (Hines, 2003), found that human resistance to change remains the top of the CRM buyers concern. Some of the users found the system intimidating.

Barton Goldberg, president of ISM, Inc., a CRM consulting firm said as many as 50% of implementation failures are directly attributable to user resistance. (Successful
According to Klaus, Wingreen and Blanton (2007), one study was found that more than half of system failed (as cited by ERP Survey Results to Need for Higher Implementation) and user resistance is one of the most important caused of this. Cooke, Dudley, Peterson and William (1998) reported that 186 companies that implemented a large system found that resistance is the second contributor to time and budget overruns and is the fourth most important barrier to implementation.

Besides of the above problems discussed, below is the analysis of three CRM software from the HCI point of view:

### Table 3.2: Comparison among CRM software

<table>
<thead>
<tr>
<th>Software/Features</th>
<th>Salesforce.com</th>
<th>Siebel CRM OnDemand</th>
<th>Microsoft CRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Tracking</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>One-click navigation to any part of application</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Modular deployment</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>List building on the fly for ease of navigation</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Simplicity of the user interface</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>User friendly</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Note: 5-Excellent, 4-Very Good, 3-moderate, 2-Poor, 1-Very Poor

A survey had been performed to view the current CRM systems. Three major CRM software vendors had been chosen which are Salesforce.com, Siebel CRM and
Microsoft CRM. A comparison was done based on HCI point of view, such as interface, user-friendliness etc. These features are then rated from 1 to 5 accordingly. Based on the Table 3.2, it can be concluded that the on-the-shelves CRM systems currently, do not adopt the usability design fully. The interface looks very promising and good, but they were not designed to meet the usability goals.
3.2 THE CALL FOR HUMAN COMPUTER INTERACTION (HCI) APPROACH IN CRM APPLICATION

The module that is focused in this proposed CRM system is the sales forecasting module. Forecasting provides tracking mechanism for the sales numbers against stock, probabilities and plans that helps the organization to fine-tune the organization-level sales process. The sales representatives' personnel can then use forecasting to meet quotas assigned to them and thus increase the company profits. Therefore the HCI approach will then be used in this CRM system in order to overcome problems faced by the current CRM software vendors in the market currently.

3.2.1 Simplify the Design of the System

User interface design is where users interact with controls or display. In many cases, good technology is not readily accepted because the product is not easy to use or efficient to use. A product's usability, acceptance, and marketability are often dependent on the user feeling that it is easy to learn and use. The simple interface increases the intuitiveness, efficiency, and comfort level with a system, thus translates into system acceptance and use. Both good technology and usability are needed for a successful system implementation. Surveys show that over 50% of the design and programming effort on projects are devoted to the user interface portion. (Myers, 1992). Therefore, by simplifying the user interface by making it more easy to use, thus will then overcome the wrong focus and no-ease-of-use issue in CRM system.
The system incorporates a highly intuitive user interface. This is done by understand the forecasting process very well, design alternatives and possibilities and breaking down process to smaller unit. One of the example is the moderation process is breaking down to 3 sub-process so that it is easier for users to handle process efficiently. New users can immediately begin to navigate around the system without formal training or reading stack of documentation products. Everything will be visible and straight forward. Only by having minimal training, they would be able to operate the system. The system will be design to eliminate the toolbar pull-downs, keyboards commands and hidden menus. Similarly, the software should allow the user to perform tasks quickly and efficiently, without sacrificing power and flexibility. This sounds simple and intuitive, and yet these considerations are so often lacking in most CRM products. A good user interface design can tell the difference between acceptance of a software product and its failure in the marketplace. If the end-users find the software to be too difficult to understand, then an otherwise excellent product could be doomed to failure. The ultimate goal is to make the system as professional-looking and easy to use as possible.

3.2.2 Involves User’s in Software Life Cycle by Adopting User Centered Approach (UCD)

As a result of reviewing over 20 articles, Ives and Olson (1984) found that participation leads to increased user acceptance and use by encouraging realistic expectations, facilitating the user's system ownership, decreasing resistance to change, and committing users to the system. According to Band.W (2005), organization which
wanted to implement CRM should follow The Bank of New York approach that makes user involvement in its project. The bank of New York had used the high user involvement approach in order to encourage 1650 of its users to use unified sales process in 32 countries. It managed to implement it successfully by getting users to be involved early in project.

By involving the end user, it will accomplish several things: weak links in processes managed to be identified, features needed by the end user on a daily basis can be understand and manage to get 'buy in' from the end user. The majority of CRM implementations fail due to lack of user buy-in. By forcing users to use the system will make them resist more. Every person wants to feel as though they have a say, thus, involving users gives the end user a voice and makes them part of the team (which also makes them each a little responsible for the success of the project). The end user is the person who will be hands on and using the system/process. CRM implementations need to be seen as a team effort throughout the organization.

The user centered approach is adopted in order to get high involvement in the system life cycle. Benefits of the user centered approach are mainly related to time and cost saving during development, completeness of system functionality, repair effort saving, as well as user satisfaction (Nielsen, 1993). It is acknowledged that approximately 60-80% of interaction difficulties, including lack of facilities and usability problems, are due to poor or inadequate requirement specifications.
Users should involve in the beginning of the process until the system deployment process. The system is designed to iterate throughout the life cycle to avoid any serious mistake and to re-save the implementation time. Table 3.3 shows the degree of user involvement in the proposed system in each phases of software life cycle.

**Table 3.3: User involvement in system life cycle**

<table>
<thead>
<tr>
<th>System life cycle</th>
<th>User involvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement gathering</td>
<td>Interview</td>
</tr>
<tr>
<td></td>
<td>Naturalistic observation</td>
</tr>
<tr>
<td></td>
<td>Studying of documentation</td>
</tr>
<tr>
<td>Analysis</td>
<td>Walk through of use case</td>
</tr>
<tr>
<td></td>
<td>User feedback</td>
</tr>
<tr>
<td>Design</td>
<td>Low fidelity prototyping</td>
</tr>
<tr>
<td></td>
<td>High fidelity prototyping</td>
</tr>
<tr>
<td></td>
<td>Prototyping evaluation</td>
</tr>
<tr>
<td>System Testing</td>
<td>Develop test plan</td>
</tr>
<tr>
<td></td>
<td>System testing</td>
</tr>
<tr>
<td></td>
<td>System evaluation</td>
</tr>
<tr>
<td>System Implementation</td>
<td>User survey to get user feedback</td>
</tr>
<tr>
<td></td>
<td>System training</td>
</tr>
</tbody>
</table>

By using user centered approach (UCD) in which users are getting involved in every phase of the software development cycle, thus will be able to address the issue of no user involvement and user resistance which had been highlighted in the Issues in CRM System.

### 3.2.3 Using Hierarchical Task Analysis (HTA)
Hierarchical Task Analysis (HTA) was originally designed to identify training needs (as cited by Roger, Y., Sharp, H. & Preece, J., 2002). It basically involves the breaking of task to subtasks. These tasks are then grouped together as plans which specify how the tasks might be performed in an actual situation. In order to come out with the HTA for the sales forecast in this project, an observation on how users performed their tasks had been done. The task decomposition of the forecasting had been carried out in the following stages:

a. Perform the task of sales forecasting.

b. Breakdown the forecasting process to subtasks.

c. Draw the forecast subtasks in layered diagram to make sure the subtask had been completed.

d. Decide which details of the subtasks need to be decomposed. This is to make sure that all the subtasks decomposition had been treated consistently.

e. Make sure the decomposition had been numbered accordingly.

f. To present the HTA to the user who are not involved in the task decomposition but know the tasks well to check for consistency and accuracy.

Below are the HTA for sales forecasting:

0 In order to produce forecast

1. Filter data

   1.1 Summarize data

2. View data

   2.1 Filter data
2.2 View graph

2.3 Spot popular item

3. Adjust forecast

3.1 Considers seasonal factors

3.2 Adjust forecast

4. Production acceptance

4.1 Accept adjusted forecast

Figure 3.1: Graphical representation of HTA of forecasting based on case study

With the use of HTA in the forecasting process will then help to address the issue of not understanding from business point of view. This was done by observing how the users who actual perform the actual forecasting process did it in the real-world.
3.2.4 Use of Metaphor

Metaphor is very popular approach to user interface design. Apple and Microsoft strongly recommended to use the metaphor.

*You can take advantage of people’s knowledge of the world around them by using metaphors to convey concepts and features in your application. Use metaphors involving concrete, familiar ideas and make the metaphors plain so that users have a set of expectations to apply to computer environments* (Apple Computer, Inc Staff 1992, as cited by Barr, P., Khaled, R., Noble, J. & Biddle, R., 2005).

![Figure 3.2: A Taxonomy of user interface metaphor](image)

Figure 3.2: A Taxonomy of user interface metaphor

(Barr, P., Khaled, R., Noble, J. & Biddle, R., 2005)
According to Barr, P et al (2005), there are 3 types of user interface metaphor:

a. Orientational metaphor

It is the metaphor that maps an interface concept to spatial concept such as up, down, left or right. In the example of up and down, the metaphoric association of ‘up’ with more and ‘down’ with less.

b. Ontological metaphor

The ontological user-interface metaphor is the one that identifies a system concept with a basic category of existence in the physical world. It allows user to use their basic knowledge of how the actual world functions to negotiate with the user interface.

c. Structural metaphor

A structural metaphor is the one that identifies an abstract system concept with a detailed world concept or objects.

Different type of metaphor described in the above figure will be use in this project. Since Microsoft Office is used widely in the case study company, these icons that associate the certain functionality in the system are associated with the icons used in
Microsoft Office. The proposed system is designed with ease of use in which icons associated to the operation is the same icons used in Microsoft applications. Thus, by looking at the icons, users will definitely know the operations that it should performed, without having to go to serious round of training. Thus, its believed to be able to help users to map the unfamiliar knowledge to familiar knowledge.
Table 3.4: The usage of metaphor in the proposed CRM system

<table>
<thead>
<tr>
<th>Metaphors</th>
<th>Usage in the system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orientational</td>
<td>The using of up, down in the date calendar. The ‘up’ indicates adding the date by 1 day and ‘down’ indicated deducting the date by 1 day. Allows direct manipulation that allows users to manipulate elements, such as mouse, of the user interface. Users will be able to scroll using the scroll bar rather than keyboard-driven command. But at the same time, it also have methods to interact with keyboard-only to cater for advance users to perform tasks more quickly, such as a shortcut key, Control-S for saving functionality.</td>
</tr>
<tr>
<td>Ontological</td>
<td>Icons used in the main functionality such as New functionality is associates with blank document, Print functionality is associates with printer icon. Besides that, all these icons are used in Microsoft Office that looked similar for users are frequent users of Microsoft Office</td>
</tr>
</tbody>
</table>
3.2.5 Identify Type of Users

According to Jurison, J (2000), perceptions of technology and adoption rates vary among type of users. Thus, there is a need to understand type of CRM users and their characteristic and behavior when performing their tasks. In the Human Computer Literature, one is usually advised to identify users based on certain user characteristic (Kujala & Kauppinen, 2004). According to Shneiderman (2005), all design should begin with understanding of the intended users including population profiles including gender, age, physical abilities, and education and so on. The participating users must be selected for field’s studies, testing, user acceptance testing and other kinds of collaboration. Furthermore, the involved users should represent the intended users of the system as closely as possible.

Thus, the user characteristic and behavior of the project will be identified. This is done using method developed by Von Hippel (1989). The idea is to conduct and interview the most advanced users in the field of interest. The lead user method had been used in uncovering innovative product ideas. The idea is that lead users face problems years before the other users encounter them, and this the lead user had already found solutions to their problems. Olson and Bakke (2001) had review experiences of using the lead user method in three published cases. Their general conclusion from the cases was that the method improved teamwork and was significantly faster and cheaper at developing new product concepts. Table 3.5 shows the user group identified in the
proposed system and their identified tasks. **Figure 3.3** shows what these user groups wanted from the system.

**Table 3.5: User group table for the proposed system**

<table>
<thead>
<tr>
<th>User group</th>
<th>Task</th>
<th>Number of users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing planner</td>
<td>View forecast</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consolidate forecast</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consolidate report</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Moderate forecast</td>
<td>5</td>
</tr>
<tr>
<td>Factory planner</td>
<td>Schedule factory plan</td>
<td>1</td>
</tr>
<tr>
<td>System administrator</td>
<td>Setup master list</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 3.3: User groups for the proposed system**
3.3 OTHER FEATURES AND CONCEPTS

3.3.1 Information Visualization

This area focuses on graphical mechanisms designed to show the structure of information and improve the cost structure of access. As business decision makers, data is used to define opportunities, trends, and areas of concern in the respective businesses. Most of the data are in the form of tabular reports and it is found challenging to quickly and effectively absorb the information, spot patterns, identify aberrations, and see hidden relationships. A picture is often worth a thousand words or in this case thousands of rows of data.

Data visualization tools can transform the large quantities of complex data into a meaningful visual representation that incorporates science behind the human perception and cognition. Data visualization applications render large quantities of data in a form of basic charts, graphical indicators, scoreboards, dashboards, advances visualizations, animations, and virtual reality environments. By automating the basic aspects of data analysis, the data visualization tools enable the information users to identify trends and patterns in data that are often not apparent in traditional tabular representations. According to Shneiderman and Plaisant (2005), data type by task taxonomy includes 7 basic data types and 7 basic tasks. The basic data are:

i. Data type
a. 1D Linear data - linear data types are one dimensional which include program source code, textual documents, dictionaries, list of names that can be organized in a sequential manner.

b. 2D map data - 2D map data includes geographic maps, floor plans and newspaper layouts. Each item has task-domain attributes and interface-domain features.

c. 3D world data – real world have complex relationship with other items. Chemical structure modeling, medical imaging are designed to handle the complex three dimensional relationships with items.

d. Multidimensional data - most relational and statistical database contents can be manipulated as multidimensional data, which items in n attributes become points in an n-dimensional space.

e. Temporal data - the distinctions of temporal data are that items have a start and finish time and the items may overlap. Frequent task include finding all the events before, after or during some time period or moment and in some cases, comparing periodical situations, plus the seven basic tasks.
f. Tree data - tree data are collections of items which items have a link to one parent item. Items and links between parent and child can have multiple attributes.

g. Network data - when relationships among items cannot be captures conveniently with a tree structure, then these items are linked to an arbitrary number of other items in a network.

The categorization is helpful to describe the visualization that had been developed and to characterize the classes of problem which user had encountered. The 7 basic tasks are as the following (Shneiderman & Plaisant, 2005):

ii. Task

a. Overview task - Users can gain an overview of the entire collection. It includes zoomed-out views of each data type which allows users to see the entire collection plus the adjoining detail view.

b. Zoom task - it enables users to zoom in items of interest. Users can zoom on one dimension at a time by moving the zoom bar controls or adjusting the size of the field of view box.
c. Filter task - by using this approach, users can filter out uninteresting items. When users control the contents of display, then they will be able to quickly focus on their interest by eliminating those unwanted items.

d. Details-on-demand task - users can select item or group in order to get the details. Once the collection had been trimmed to a few dozen items, then it should be easier to browse the details on the individual items.

e. Relate task – users would be able to relate items or groups within the collection. It can make use of the human perceptual ability for visual information. Within the visual displays, there are opportunities to show the relationship by proximity, connected lines or colour coding.

f. History task – users would be able to keep history of action list in order to support replay, undo and refinement.

g. Extract task – Extraction of sub-collections and the query parameters are allowed. Once the user obtained the items, it is useful for them to be extract that set and save them, sending it through email.

Based on the case study conducted on CST, since the user’s tasks are more on looking on overview of the collection of tyre size and filtering the uninteresting or
unpopular size, therefore 1D liner data is chosen as it will list down all the tyre size of
highest demand. The attribute had been identified as tyre size and each attribute is colour-
coded. The attribute is ordered based on the popularity of the tyre size and the number of
the most popular items to be displayed will be determine by the user. If the user wants 50
popular items, therefore system will list down the most 50 highest figures submitted by
the sales office, with coded colours on the chart.

The data will be represented in a form of bar charts. The use of pie charts and bar
charts help the user to quickly identify general upwards and downwards trends as well as
identify groups within the data that are performing differently than other groups. By
using the chart, user is able to see how data relates to the other data instead of looking at
tabular list of numbers. Beside this, it also allows the user to spend more time resolving
the problem or pursuing opportunity and less time trying to discover or define them. By
generating the chart automatically, the time that the user has to spend to create the chart
manually will be reduced, such as the one in spreadsheets.

3.3.2 Case Base Reasoning (CBR)

Case Base Reasoning or CBR is an intelligent system method that enables users to
increase efficiently and reduce cost by substantially automating processes. It is one of the
most promising technologies for building intelligence into the computers. CBR solves
new problems by adapting previously successful solutions to similar problems.
a) CBR Cycle

The processes involved in CBR can be represented by a schematic cycle (refer Figure 3.4). Aamodt and Plaza (1994) had described CBR typically as cyclical processes comprising of the four REs (as cited by Watson & Maris, n.d.)

- REtrieve the most similar case(s)
- REuse the case(s) to attempt to solve the problem
- REvise the proposed solution if necessary
- RETain the new solution as a part of a new case

![Figure 3.4: The CBR Cycle (Aamodt and Plaza, 1994)]
A new problem is matched against the case in the case base and one or more similar cases are retrieved. A solution suggested by the matching case is then reused and tested for success. Unless the retrieved result is a close match, the solution will probably have to be revisited to produce a new case that can be retrieved. This cycle rarely occurs without the human intervention. This therefore encourages human collaboration in decision support.

b) Case Representation

A case is a contextualized piece of knowledge representing an experience. It contains the past lesson that is the content of the case and the context in which the lesson can be used (Alterman, 1989). Typically a case comprises:

- the problem that describes the state of the world when the case occurred.
- the solution which states the derived solution to that problem, and/or
- the outcome which describe the state of the world after the case occurred.

Cases which comprise problems and their solutions can be used to derive solutions to new problems, as in CASEY (Koton, 1989). Whereas cases comprises of problems and outcomes can be used to evaluate new situations. If, in addition, such cases contain solutions they can be used to evaluate the outcome of proposed solutions and prevent potential problems as in MEDIATOR (Simpson, 1985). Cases can be represented in a variety of forms using the full range of AI representational formalisms including frames, objects, predicates, semantic nets and rules - the frame/object representation currently being used by the majority of CBR software.
c) Indexing

Case indexing involves assigning indices to cases to facilitate their retrieval. Several guidelines on indexing have been proposed by CBR researchers (Birnbaum & Collings, 1989). Indices should as the following:

- be predictive,
- address the purposes the case will be used for,
- be abstract enough to allow for widening the future use of the case-base, and
- be concrete enough to be recognized in future

Both manual and automated methods have been used to select indices. Choosing indices manually involves deciding a case’s purpose with respect to the aims of the reasoner and deciding under what circumstances the case will be useful.

d) Storage

Case storage is an important aspect in designing efficient CBR systems in that, it should reflect the conceptual view of what is represented in the case and take into account the indices that characterize the case. The case-base should be organized into a manageable structure that supports efficient search and retrieval methods. A balance has to be found between storing methods that preserve the semantic richness of cases and their indices and methods that simplify the access and retrieval of relevant cases. These methods are usually referred to as case memory models. The two most influential case memory models are the *dynamic memory model* of Schank and Kolodner, and the *category-exemplar model* of Porter and Bareiss (Watson & Marir, 1994).

i. The dynamic memory model
The case memory model in this method is comprised of memory organization packets (MOP). MOPs are a form of frame and are the basic unit in dynamic memory. The case memory, in a dynamic memory model, is a hierarchical structure of episodic memory organization packets (E-MOP), also referred to as generalized episode (GEs) (Koton, 1989) developed from Schank's more general MOP theory (Schank, 1982). The basic idea is to organize specific cases which share similar properties under a more general structure (example, a generalized episode). A GE contains three different types of objects: norms, cases and indices. Norms are features common to all cases indexed under a GE. Indices are features which discriminate between a GE’s cases. An index may point to a more specific generalized episode or to a case, and is composed of an index name and an index value.

The case-memory is a discrimination network where nodes are either a GE, an index name, index value or a case. Index name-value pairs point from a GE to another GE or case. The primary role of a GE is as an indexing structure for storing, matching and retrieval of cases. During case storage when a feature (example index name and index value) of a new case matches a feature of an existing case a new GE is created. The two cases are then discriminated by indexing them under different indices below the new GE (assuming the cases are not identical). Thus, the memory is dynamic in that similar parts of two cases are dynamically generalized into a new GE, the cases being indexed under the GE by their differences.
However, this process can lead to an explosive growth in the number of indices as case numbers increase. So for practical purposes most CBR systems using this method limit the number of permissible indices to a limited vocabulary.

ii. The category-exemplar model

This model organizes cases based on the view that the real world should be defined extensionally with cases being referred to as exemplars (Porter & Bareiss, 1986). The case memory is a network structure of categories, semantic relations, cases and index pointers. Each case is associated with a category. Different case features are assigned different importance in describing a case's membership to a category. Three types of indices are provided, which may point to a case or a category:

- feature links that point from problem descriptors (features) to a case or category,
- case links that point from categories to its associated cases, and
- difference links pointing from categories to the neighboring cases that only differ in a small number of features.

A feature is described by a name-value pair. A category’s exemplars are stored according to their degree of prototypicality to the category. Within this memory organization, the categories are inter-linked within a semantic network containing the features and intermediate states referred to by other terms. This network represents a background of general domain knowledge that enables explanatory support to some CBR tasks. A new case is stored by searching for a matching case and by establishing the
relevant feature indices. If a case is found with only minor differences to the new case, the new case may not be retained, or the two cases may be merged.

e) Retrieval

Given a description of a problem, a retrieval algorithm, using the indices in the case-memory, should retrieve the most similar cases to the current problem or situation. The retrieval algorithm relies on the indices and the organization of the memory to direct the search to potentially useful cases.

The issue of choosing the best matching case has been addressed by research into analogy (Falkenheiner, Forbus & Gentner, 1986). This approach involves using heuristics to constrain and direct the search. Several algorithms have been implemented to retrieve appropriate cases, for example: serial search (Navichandra, 1991), hierarchical search (Maher & Zhang, 1991) and simulated parallel search (Domeshek, 1993).

Case-based reasoning will be ready for large scale problems only when retrieval algorithms are efficient at handling thousands of cases. Unlike database searches that target a specific value in a record, retrieval of cases from the case-base must be equipped with heuristics that perform partial matches, since in general there is no existing case that exactly matches the new case.
Among well known methods for case retrieval is: nearest neighbor, induction, knowledge guided induction and template retrieval. These methods can be used alone or combined into hybrid retrieval strategies.

i. Nearest neighbour

This approach involves the assessment of similarity between stored cases and the new input case, based on matching a weighted sum of features. The biggest problem here is to determine the weights of the features. The limitation of this approach includes problems in converging on the correct solution and retrieval times. In general the use of this method leads to the retrieval time increasing linearly with the number of cases. Therefore this approach is more effective when the case base is relatively small. Several CBR implementations have used this method to retrieve matching cases, for example: BROADWAY (Skalk, 1992) for selection of car models, the Compaq SMART System (Acorn & Walden, 1992) for a customer product support help desk, and ANON (Owens, 1993) for situation assessment in plan failure.

A typical algorithm for calculating nearest neighbour matching is the one used by Cognitive Systems ReMind software reported in Kolodner (Kolodner, 1993) where w is the importance weighting of a feature (or slot), sim is the similarity function, and fI and fR are the values for feature i in the input and retrieved cases respectively.
ii. Induction

Induction algorithms (Quinlan, 1979) determine which features do the best job in discriminating cases, and generate a decision tree type structure to organize the cases in memory. This approach is useful when a single case feature is required as a solution, and where that case feature is dependent upon others.

iii. Knowledge guided induction

This method applies knowledge to the induction process by manually identifying case features that are known or thought to affect the primary case feature. This approach is frequently used in conjunction with other techniques, because the explanatory knowledge is not always readily available for large case bases.
iv. Template retrieval

Similar to SQL-like queries, template retrieval returns all cases that fit within certain parameters. This technique is often used before other techniques, such as nearest neighbour, to limit the search space to a relevant section of the case-base.

In this project, a nearest neighbor algorithm is chosen to retrieve the solution for the moderation process.

f) Adaptation

Once a matching case is retrieved a CBR system should adapt the solution stored in the retrieved case to the needs of the current case. Adaptation looks for prominent differences between the retrieved case and the current case and then applies formulae or rules that take those differences into account when suggesting a solution. In general, there are two kinds of adaptation in CBR:

- Structural adaptation, in which adaptation rules are applied directly to the solution stored in cases (Kolodner, 1993). This kind of adaptation is used in JUDGE (Bain, 1986) and CHEF (Hammond, 1986).
- Derivational adaptation that reuses the algorithms, methods or rules that generated the original solution to produce a new solution to the current problem. In this method the planning sequence that constructed that original solution must be stored in memory
along with the solution as in MEDIATOR (Simpson, 1985). Derivational adaptation, sometimes referred to a re-instantiation, can only be used for cases that are well understood.

An ideal set of adaptation rules must be strong enough to generate complete solutions from scratch, and an efficient CBR system may need both structural adaptation rules to adapt poorly understood solutions and derivational mechanisms to adapt solutions of cases that are well understood.

A basic premise in CBR is that many problems that decision makers encounter are not unique, but rather they are variations of a problem type. It is often more efficient to solve a problem by starting with the solution to a previous similar problem than it is to generate the entire solution again from first principles. In fact, experts have been observed to reason by analogy to prior cases. In solving a current problem, a case-based reasoner (whether it is a human or a computational model) recalls a similar past case and its solution. The reasoner then adapts the successful solution of the recalled case to adjust for any differences between the current case and the recalled case. Finally, the CBR stores the solution to the current case along with feedback about the outcome so that it can be used in solving future problems.
There are a number of advantages to using CBR. In domains that lack a strong domain theory, model-based reasoners are not practical. When the relationship between the cases attributes and the solution or outcome is not understood well enough to represent it in rules, or when the ratio of cases that are "exceptions to the rule" is high, rule-based systems become impractical. CBR is especially useful in such situations because it models the exceptions and the novel cases.

CBR is also useful in explaining or justifying a solution. When the domain theory is weak, it is difficult to justify or explain a position based on first principles. Drawing the analogy to a similar prior case may be more persuasive than a model-based argument. CBR similarly supports learning in domains that lack a strong domain model. For example, the case method of teaching is used in law schools and business schools. Users of a CBR may find it more instructive to see examples of previous cases than to see rules.
3.4 CONCLUSION

This chapter had discussed the issues and fallback of the current CRM applications. It highlights the importance of having Human Computer Interaction (HCI) approach when designing a CRM application as it had stressed not only for high usability, but also new interaction techniques for supporting user tasks, providing better access to information, and creating more powerful forms of communication. By incorporating the User Centered Design (UCD) approach in the CRM system, thus will be the start of the acceptance of the project implementation by the user community since it will be seen as coming from their own. Upon understanding the user and their needs is the transferring it to the designing of the user interface.
CHAPTER 4

RESEARCH METHODOLOGY

4.0 INTRODUCTION

The research methodology should discuss the problems that were anticipated and explain the steps taken to prevent them from occurring, and the problems that did occur and the ways their impact was minimized. The research is done on qualitative point of view as it more concerned with understanding why people behave as they do, to know on their knowledge, attitudes, beliefs, fears etc. In this project, we would want to know why people behave in certain behaviors, so that an interactive CRM system can be developed to help the users achieve their task effectively and efficiently. Below is the research methodology flow that is used in this project.
Figure 4.1: Research Methodology Flow
4.1 RESEARCH METHODOLOGY

4.1.1 Single In-Depth Case Study

Single In-Depth Case Study is chosen as a research method. A case study of a multinational company had been chosen in this research, in which the background, current conditions and environmental interactions are observed, recorded and analyzed for stages of patterns in relation to internal and external influences. The steps involves in this single in-depth case study are:

i. Implementation of User Centered Design and HCI approach in CRM system.

ii. Implementation of Case Base Reasoning (CBR) in automating the decision making process in CRM system.

4.1.2 Literature Review

A few CRM systems that available in the market had been reviewed in order to know the current features and what are the strength and weakness in the system from the Human Computer Interaction (HCI) point of view. Besides that, different type of orientation approach had been reviewed to examine the customer data. All the information is collected through the online system, mainly from:

i. Online journal.

ii. HCI and CRM Books.

iii. CRM resource websites such as business websites, case study, articles, white papers, proceedings and conference papers.
iv. HCI resource websites such as university websites, case study, articles, journal, white papers, proceedings and conference papers.

4.1.3 Data Gathering

The main goal of data gathering technique is to collect sufficient, relevant and appropriate data so that a set of complete requirement can be produced. In the data gathering technique, we need to know the goal of the tasks and the current tasks that user normally perform. According to Preece, Rogers and Sharp (2002), there are five data-gathering technique used in the requirement study. These techniques are flexible and can be combined and extended in many ways so that the data-gathering technique can be varied and managed to give full leverage on understanding of different requirement. The techniques are questionnaires, interviews, focus group and workshop, naturalistic observation and studying documentation.

Table 4.1: Overview of data-gathering technique used in requirements activity

<table>
<thead>
<tr>
<th>Technique</th>
<th>Good for</th>
<th>Type of data</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Questionnaires</td>
<td>Answering specific question</td>
<td>Quantitative and qualitative data</td>
<td>Can reach many people with low resource</td>
<td>The design is crucial. Response rate may be low. Responses may not be want you want</td>
</tr>
<tr>
<td>Interviews</td>
<td>Exploring issue</td>
<td>Some quantitative but mostly qualitative data</td>
<td>Interviewer can guide interviewee of necessary. Encourages contact between developers and users</td>
<td>Time consuming. Artificial environment may intimidate interviewee</td>
</tr>
<tr>
<td>Focus group and workshop</td>
<td>Collecting multiple viewpoints</td>
<td>Some quantitative but mostly qualitative data</td>
<td>Highlights areas of consensus and conflict. Encourages contact between developers and users</td>
<td>Possibility of dominant characters</td>
</tr>
</tbody>
</table>
Olson and Moran (1996) suggest that choosing between data-gathering techniques should base on two factors, which are:

i. Nature of the data gathering itself

ii. The task to be studied.

A few techniques are combined, which are interviews, naturalistic observation and studying of documentation when gathering the requirements from the users. The methods had been chosen due to the reason that there is need to really understand the natural activity performed when users are carrying their task, exploring issues related to the process and learning the current standards in the process. Type of the interview conducted is the unstructured interview, in which it is more like a conversation that stressed on a particular topic. This type of interview generates rich data because the interviewee normally mentions things that the interviewer may not have considered and can be expanded more.
4.2 SOFTWARE DEVELOPMENT

Software development is achieved through series of activities that lead to the development of the system. The software development can be divided into four phases which are as the following:

4.2.1 Capturing System Requirement

The Use Case Driven methodology had been one of the popular methods to capture the requirement analysis. Besides, the use of Use Case had been proven as effective mechanism to capture the requirement. It is an effective way of capturing both Business Processes and System Requirements. All the typical scenarios will be documented and use case is a technique for formalizing the capture of these scenarios. Among of the benefits of using use cases are:

i. Use case focus on users of the system and not the system itself. Thus, the system needs are brought to light early on.

ii. Use case is easy to understand by all stakeholders (including customers, executives, clerks and not only developers and testers) as it uses narrative text.

iii. Use case provides traceability throughout the system development process.

Since use case can be used to identify objects from scenarios, and identify requirements from the system, therefore it is chosen to capture the system requirement in
this project. For each module of the system, the use cases will be converted into a list of requirements for that module to be implemented successfully.

4.2.2 Analysis and Design

During the analysis phase, data are collected on the available files, decision points and transactions handled by the present manual system. Interviews, on-site observation and questionnaire are the tools used for system analysis.

All procedures and requirements must be analyzed and documented in the form of use case diagram and sequence diagram. System Analysis also includes sub-dividing of complex process involving the entire system, identification of data store and manual processes.

System design is the most crucial phase in the development of a system. Normally, the design proceeds in two stages:

i. Preliminary or general design - features of the new system are specified. The benefits of the projects will be derived. If the project is still considered to be feasible, we move to the detailed design stage.
ii. Structure or detailed design - at this stage, the design of the system becomes more structured. Input, output and processing specifications are drawn up in detail. In the design stage, the programming language and the platform in which the new system will run are also decided.

4.2.3 System Testing

Series of system testing need to be performed in order to make sure that it meets the user’s requirements. Among the testing that need to be performed are:

i. Unit testing – The testing is to test the individual components of the system.

ii. Integration testing – This testing is to test the functionality and integration between modules in the system.

iii. Acceptance testing – the testing is to receive the feedback from the users regarding the system usability and the functionality of the system. It should be able to verify whether the final deliverables meet the user’s expectations.

4.2.4 System Implementation

The system implementation is to make sure that the new system available to a prepared set of users. The deploying of the system includes executing all steps necessary to educate users on the new system including user training and providing user manual for guidelines.
4.3 SOFTWARE DEVELOPMENT METHODOLOGY

4.3.1 Introduction

   Flexibility in the requirement, analysis and design phase of the life cycle is a must. The user has to be able to change requirements and this change is reflected in the analysis, design and coding. To handle this situation a comprehensive set of case tools is needed and these tools must be good enough to accept this flexibility. The selection of an appropriate process model is basically depends on the organizational environment. Kand. F (1998) suggests that the suitable approach to system analysis, design, development and implementation is based on the relationship between the information system and its organizational environment. Based on A Survey of System Development Process Models (n.d), it was found that there are four types of the relationship, which are define as the following:

i. The unchanging environment

   The requirements are unchanging for the lifetime of the system. A high degree of accuracy is essential. In this environment, Waterfall or Spiral models would provide the completeness provided by the system.

ii. The turbulent environment

   The organization is undergoing constant changes and system requirements are always changing (“Creating and Combining models,” n.d.). Many business systems fall into this category. Successful methods would include those which incorporate rapid development such as Prototyping and the maximum use of reusable code.
iii. The uncertain environment

The requirements are unknown or uncertain. It is not possible to define the requirements accurately ahead of time because the situation is new or the system being employed highly innovative. Experimental process model which takes advantage of prototyping and rapid development are most appropriate.

iv. The adaptive environment

The environment may change in reaction to the system being developed, thus initiating a changed set of requirements. Teaching and expert system fall into this category.

Based on the observation, it was found out that the company’s environment in the use case study (Continental Sime Tyre PJ Sdn. Bhd), fell under the turbulent environment, in which the requirements are keep on changing and it is quite hard to define the requirements at the initial stage. This is because at the time the system is initiated; the management of the company had changed from local management to German’s management. Due to this fact, then a rapid development is most suitable for the environment. User Centered Design (UCD) approach with prototyping modeling is the most suitable model to be used in the system. (Details available in Chapter 2 – Literature Review)
4.3.2 Prototyping Modeling

Prototypes are useful when discussing ideas with the stakeholders. They are the communication aid among the team members and one of the effective approaches to test ideas. It can be used to suit many purpose, such as to clarify some unclear requirements or to perform user testing and evaluation.

There are two types of prototyping which are as follows:

(a). Low fidelity prototyping

Low fidelity prototyping is the one that does not look very much like the final product. It can be useful because they are simple, quick and cheap to produce. Paper prototype is chosen because the work can be done faster, better, more independent and with lower risk. The design changes are easily made with a pen/pencil. In this project, paper sketches are used because it can be produced very quickly; can be easily modified and the medium is very flexible. It is faster to come out with the sketches and if there is a change, the design can be erased and re-draw. Paper sketches are produced in the first iteration of the prototyping life cycle. This is explained in detail in Chapter 6 – Analysis and Design.

(b). High fidelity prototyping

The high fidelity prototyping is a method where the prototype used for testing the actual interface as closely as possible. Normally, software tool is used to mockup the interface. The software tool accepts the input from the keyboard or mouse like the actual
interface would do and responds to those events in the same way. In this project, the Visual Basic 6.0 is used to design the interface prototype in the second iteration upon the compilation of the feedback received from the user in the first iteration of prototyping life cycle. Visual Basic is chosen because it is known for its rapid application development whereby it can build the user interface visually instead of having to pound out all the code by hand. In Visual Basic, the controls can be dragged and dropped from a toolbox into the form and have the skeleton code written. The screen of the second iteration of prototype life cycle is explained in details in Chapter 6 – Analysis and Design.

Phase: Requirement Definition/Collection

Table 4.2: Requirement Definition Goal, Task and Result

<table>
<thead>
<tr>
<th>Goal/Task/Result</th>
<th>Definition</th>
</tr>
</thead>
</table>
| Goal             | ▪ To understand the characteristic and behavior of user  
|                  | ▪ To determine the characteristic of the user’s tasks  
|                  | ▪ To determine the objectives and constraint  
|                  | ▪ To understand the user requirements accurately  
|                  | ▪ To map the user requirements inside the system  |
| Task             | ▪ Have an interview with user to get to know them and know the process of the business process  
|                  | ▪ Define the scope of the system  
|                  | ▪ Identify the functional, non-functional and usability requirements of the system  |
| Result           | ▪ The scope of the system is limited to moderation process and the producing of the factory program  
|                  | ▪ The functional, non-functional and usability requirements had been identified as explained in Chapter 4  |
Phase: Design

**Table 4.3: Design Goal, Task and Result**

<table>
<thead>
<tr>
<th>Goal/Task/Result</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>▪ To design the interface of the screen by using the prototype</td>
</tr>
<tr>
<td></td>
<td>▪ To design and model the database using the Unified Modeling Language (UML)</td>
</tr>
<tr>
<td>Task</td>
<td>▪ To build and design the database structure</td>
</tr>
<tr>
<td></td>
<td>▪ To build use case, activity diagram and sequence diagram</td>
</tr>
<tr>
<td></td>
<td>▪ To build the interface/screen based on the database design</td>
</tr>
<tr>
<td>Result</td>
<td>▪ Drafted ERD as in the Chapter 6 – Analysis and Design</td>
</tr>
<tr>
<td></td>
<td>▪ Drafted the use case diagram, activity diagram and sequence diagram as in the Chapter 6 – Analysis and Design</td>
</tr>
<tr>
<td></td>
<td>▪ Built the interface of the system</td>
</tr>
</tbody>
</table>

Phase: Prototype Creation/Modification

**Table 4.4: Prototype Creation Goal, Task and Result**

<table>
<thead>
<tr>
<th>Goal/Task/Result</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>▪ To develop the prototype of interface designs</td>
</tr>
<tr>
<td>Task</td>
<td>▪ To develop the low fidelity prototype using paper sketches prototype</td>
</tr>
<tr>
<td></td>
<td>▪ To develop the high fidelity prototype using Visual Basic user interface</td>
</tr>
<tr>
<td></td>
<td>▪ To compiled feedback received from each iteration and incorporate them inside the prototype</td>
</tr>
<tr>
<td>Result</td>
<td>▪ Developed the prototype of interface designs to meet the user requirements</td>
</tr>
</tbody>
</table>
Phase: System Implementation

Table 4.5: System Implementation Goal, Task and Result

<table>
<thead>
<tr>
<th>Goal/Task/Result</th>
<th>Definition</th>
</tr>
</thead>
</table>
| **Goal**         | • Incorporate the HCI approach and user requirements inside the system  
                    • Construction of the modules in the system |
| **Task**         | • To transfer the design and model into the construction of the system |
| **Result**       | • All the screens in the system are coded as attached in Appendix D |

Phase: System Testing

Table 4.6: System Testing Goal, Task and Result

<table>
<thead>
<tr>
<th>Goal/Task/Result</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal</strong></td>
<td>• To test each screens in the system to make sure that it works correctly and meets the user requirements</td>
</tr>
</tbody>
</table>
| **Task**         | • To test each screens in the system to make sure that it works correctly and meets the user requirements. The screens are login, company selection, contributor, feature, user, user case, moderation, factory program and change company screens. This is done by following the test plan as in Chapter 8 – System Testing  
                    • To test the integration of system to ensure that data from master screens flow correctly to the transaction screens |
| **Result**       | • The system is free errors and working correctly as per the user requirement |

All the questions asked during the data collection had been compiled and attached in Appendix D – Interview Questionnaires
4.4 CONCLUSION

This chapter explains on research methodology that describes the methods and approaches used throughout the research of this project. It explains in brief each phases in the research methodology, as well as the software development phases. Besides that, it also explains on the software development methodology used in this project.
CHAPTER 5

REQUIREMENT ANALYSIS

5.0 INTRODUCTION

Requirement analysis is a stage in which all the client requirements are gathered. Requirement analysis is the first technical step in the software process. It is at this point that a general statement of software scope is refined into a concrete specification that becomes the foundation for all software engineering activities that follow. Analysis must focus on information, functional, and behavioral domains of a problem. It is essential to establish correct requirements and specifications early in the development process to prevent errors later on in the system life cycle.

This chapter elaborates on the system’s requirements which were gathered on the basis of information provided by the users in the form of documents, existing manual system and process specs and on-site analysis interviews with end-users. All these are documented using the use cases that are capable to capture the system functionality from user’s perspective. Based on the list of modules identified in the system, each use case for each module will be converted into a list of requirements.

5.1 GENERAL ASSUMPTION

The following assumptions have been made in the elicitation of the requirements:
i. The users of the system already understood well how to do manual decision making in moderation process.

ii. The users of the system are familiar with the Microsoft tools.

5.2 USER CHARACTERISTIC

Based on the observation (as explained in details in Chapter 2 – Literature Review), the users of the system can be classified as knowledgeable intermittent user in which they already know concept of computer and have ideas on tasks to be performed. The users who are going to use computers are marketing planner and factory planner who are already familiar with their manual task and had been using Microsoft products (such as Excel and Word) when performing their tasks.
5.3 LIST OF MODULES

Below is the list of modules available in the system:

i. Sign Up module.

ii. Access Authentication module.

iii. Consolidate Forecast module.

iv. Populate Graph module.

v. Moderate Forecast module.

vi. Schedule Factory Plan module.

vii. Maintain Main Contributor module

viii. Maintain Feature module

ix. Change Password
5.3.1 Signup Module

The system administrator needs to register the personal details in the sign up module before any of the users can start using the system. The details such as user’s employee code, name and network id must be provided in order to determine the accuracy of the data. The creation of user id that is based on network id must be unique.

Figure 5.1 illustrates the use case diagram for Sign-up Module.

Figure 5.1: Use Case Diagram for the Sign-Up Module
Based on the above use case and its description, below is the requirements derived, as shown in Table 5.1:

Table 5.1: The System Requirements for the Sign-up Module

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a registration interface to allow system administrator to register the personal details.</td>
<td>CRM-Mod01-01</td>
</tr>
<tr>
<td>2.</td>
<td>Each registration screen of the system will consist of a user id, employee name, network id and password.</td>
<td>CRM-Mod01-02</td>
</tr>
<tr>
<td>3.</td>
<td>The system will ensure the user id created is uniquely named after the network id.</td>
<td>CRM-Mod01-03</td>
</tr>
<tr>
<td>5.</td>
<td>The system will verify the input data before storing the data in the database by triggering an alert to the user if the value is invalid.</td>
<td>CRM-Mod01-04</td>
</tr>
</tbody>
</table>

5.3.2 Access Authenticate Module

Access authentication module authenticates the accesses and requests users. During the authenticate process, only an authorized user is allowed to login inside the system. It requires a valid password associated with the user id in order to access the data.
Only user with access is allowed to view their screen. **Figure 5.2** illustrates the use case diagram for the access authentication module.

![Use Case Diagram for Access Authenticate Module](image)

**Figure 5.2: Use Case Diagram for Access Authenticate Module**

Three requirements must be fulfilled to implement the Access Authentication module, as described in **Table 5.2** below:

**Table 5.2: The Requirements for the Access Authentication Module**

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a login interface that consists of the username, password and database fields.</td>
<td>CRM-Mod02-01</td>
</tr>
<tr>
<td>2.</td>
<td>Both inputs shall be verified first by the system in order to</td>
<td>CRM-Mod02-02</td>
</tr>
</tbody>
</table>
authenticate the user before entering the system.

3. The system will verify and authenticate the screen accesses made by the authorized user.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CRM-Mod02-03</td>
<td></td>
</tr>
</tbody>
</table>

### 5.3.3 Consolidate Forecast Module

Consolidate Forecast module allows the marketing planner to summarize the forecast details which were input from the sales representative. During this process, users are allowed to select sales office in order to filter the forecast based on certain sales office. But at the same time, system allows the flexibility to users to display all the sales office if wanted to do so.

![Consolidate Forecast Diagram](image)

**Figure 5.3: Use Case Diagram for Consolidate Forecast**

Eight requirements must be fulfilled to implement the Consolidate Forecast module, as described in **Table 5.3** below:
### Table 5.3: The Requirement for the Consolidate Forecast

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a consolidate forecast interface that consists of the item code, item name, forecast month and the number of item forecast for the month.</td>
<td>CRM-Mod03-01</td>
</tr>
<tr>
<td>2.</td>
<td>The system will listed all the sales office available in the company</td>
<td>CRM-Mod03-02</td>
</tr>
<tr>
<td>3.</td>
<td>The system will allow the selection of the sales office by the user</td>
<td>CRM-Mod03-03</td>
</tr>
<tr>
<td>4.</td>
<td>The system will display the consolidated forecast figures based on the next six month.</td>
<td>CRM-Mod03-04</td>
</tr>
<tr>
<td>5.</td>
<td>The system will allow user to export the data to the Microsoft spreadsheet to allow them to manipulate data inside that tool.</td>
<td>CRM-Mod03-05</td>
</tr>
<tr>
<td>6.</td>
<td>The system will allow the details to be printed.</td>
<td>CRM-Mod03-06</td>
</tr>
<tr>
<td>7.</td>
<td>The system should prompt users for confirmation each time user wanted to perform the task</td>
<td>CRM-Mod03-07</td>
</tr>
<tr>
<td>8.</td>
<td>The system, as much as possible should provide informative content to user.</td>
<td>CRM-Mod03-08</td>
</tr>
</tbody>
</table>
5.3.4 Populate Graph Module

Populate graph enables user to visualize the item based on the highest number of items forecasted. During this process, users are allowed to select product group in order to filter the forecast based on certain product group. But at the same time, system allows the flexibility to users to display the entire product group if wanted to do so.

![Figure 5.4: Use Case Diagram for Populate Graph](image)

Five requirements must be fulfilled to implement the Populate Graph module, as described in Table 5.4 below:

Table 5.4: The Requirement for the Populate Graph

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a populate graph interface that consists of the item code and the number of item forecast for</td>
<td>CRM-Mod04-01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>2.</strong></td>
<td>The system will allow user to set the selection by allowing them to choose type of graph that will be display, the forecast month and the top number of item that will be display.</td>
<td>CRM-Mod04-02</td>
</tr>
<tr>
<td><strong>3.</strong></td>
<td>The system shall gives flexibility to define the most popular items in the range of 1 to 100.</td>
<td>CRM-Mod04-03</td>
</tr>
<tr>
<td><strong>4.</strong></td>
<td>The item shall be coded with different colours to make it easier to visualize.</td>
<td>CRM-Mod04-04</td>
</tr>
<tr>
<td><strong>5.</strong></td>
<td>The system shall sort the item based on the highest number of item</td>
<td>CRM-Mod04-05</td>
</tr>
<tr>
<td><strong>6.</strong></td>
<td>The system will allow the details to be printed.</td>
<td>CRM-Mod04-06</td>
</tr>
<tr>
<td><strong>7.</strong></td>
<td>The system should prompt users for confirmation each time user wanted to perform the task.</td>
<td>CRM-Mod04-08</td>
</tr>
<tr>
<td><strong>8.</strong></td>
<td>The system shall allow the graph to be enlarged.</td>
<td>CRM-Mod04-09</td>
</tr>
</tbody>
</table>

### 5.3.5 Moderate Forecast Module

Moderate Forecast module is to allow the marketing planner to moderate the forecast based on factors that had been defined in the Main Contributor module. During this process, users are allowed to select product group in order to filter the forecast based on certain product group. But at the same time, system allows the flexibility to users to display the entire product group if wanted to do so.
Eleven requirements must be fulfilled to implement the Moderate Forecast module, as described in Table 5.5 below:

**Table 5.5: The Requirement for the Moderate Forecast**

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a moderate forecast interface that consists of the item code, item name, factory code, available stock, forecast number and the number of item forecast based on market.</td>
<td>CRM-Mod05-01</td>
</tr>
<tr>
<td>2.</td>
<td>The system will highlight the number of forecast that is more than 1000, with yellow colour. This is for easier identification</td>
<td>CRM-Mod05-02</td>
</tr>
</tbody>
</table>
as these items are popular items and demanded more by the customers.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.</td>
<td>The system shall sort the details by item code.</td>
</tr>
<tr>
<td>4.</td>
<td>The system shall allow the marketing planner to modify the number of forecast by each market.</td>
</tr>
<tr>
<td>5.</td>
<td>The system should auto-calculate the total number of item upon changes made by the marketing planner</td>
</tr>
<tr>
<td>6.</td>
<td>The system will allow user to export the data to the Microsoft spreadsheet to allow them to manipulate data using the tool.</td>
</tr>
<tr>
<td>7.</td>
<td>The system will allow the details to be printed.</td>
</tr>
<tr>
<td>8.</td>
<td>The system should allow user to use the same mode (new, save and delete) which is available in the other interfaces.</td>
</tr>
<tr>
<td>9.</td>
<td>The system should prompt users for confirmation each time user wanted to perform the task</td>
</tr>
<tr>
<td>10.</td>
<td>The system, as much as possible should provide informative content to user.</td>
</tr>
</tbody>
</table>

### 5.3.6 Schedule Factory Plan Module

Schedule Factory Plan is a module to allow the factory planner to schedule the factory plan/program based on the moderated marketing figures.
Seven requirements must be fulfilled to implement the Schedule Factory Plan module, as described in Table 5.6 below:

### Table 5.6: The Requirement for the Schedule Factory Plan

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a schedule factory plan interface that consists of rim size, the item code, item name, factory code, the number of moderated item based on market and scheduled items based on weeks.</td>
<td>CRM-Mod06-01</td>
</tr>
<tr>
<td>2.</td>
<td>The system will display the number of moderated items based on market.</td>
<td>CRM-Mod06-02</td>
</tr>
<tr>
<td>3.</td>
<td>The system shall allow the factory planner to edit the moderated item on weekly basis</td>
<td>CRM-Mod06-03</td>
</tr>
<tr>
<td></td>
<td>The system should allow user to use the same mode (new, save and delete) which is available in the other interfaces.</td>
<td>CRM-Mod06-04</td>
</tr>
<tr>
<td>---</td>
<td>---------------------------------------------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>5.</td>
<td>The system should prompt users for confirmation each time user wanted to perform the task</td>
<td>CRM-Mod06-05</td>
</tr>
<tr>
<td>6.</td>
<td>The system, as much as possible should provide informative content to user.</td>
<td>CRM-Mod06-06</td>
</tr>
<tr>
<td>7.</td>
<td>The system will verify the input data before to storing the data in the database by triggering an alert to the user if the value is invalid.</td>
<td>CRM-Mod06-07</td>
</tr>
</tbody>
</table>
5.3.7 Maintain Main Contributor Module

Main Contributor modules allows user to input the factors that contribute to decision making in moderation process.

![Use Case Diagram for Maintain Main Contributor](image)

**Figure 5.7: Use Case Diagram for Maintain Main Contributor**

Nine requirements must be fulfilled to implement the Maintain Main Contributor module, as described in Table 5.7 below:

**Table 5.7: The Requirement for the Maintain Main Contributor**

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a main contributor interface that consists of contributor code, contributor name and weight.</td>
<td>CRM-Mod07-01</td>
</tr>
<tr>
<td>2.</td>
<td>The system shall allow user to sort the contributor listing based on their preference.</td>
<td>CRM-Mod07-02</td>
</tr>
<tr>
<td>3.</td>
<td>The system should only allow numeric values in Weight field</td>
<td>CRM-Mod07-03</td>
</tr>
<tr>
<td>4.</td>
<td>The system should allow user to use the same mode (new, save</td>
<td>CRM-Mod07-04</td>
</tr>
</tbody>
</table>
and delete) which is available in the other interfaces.

5. The system should prompt users for confirmation each time user wanted to perform the task

6. The system, as much as possible should provide informative content to user.

7. The system should allow the input of data to be done rapidly with the using of keyboard by following the sequence of field on the screen

8. The values for Weight should ranging from 1 to 10

9. The system will verify the input data before storing the data in the database by triggering an alert to the user if the value is invalid.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td>The system should prompt users for confirmation each time user wanted to perform the task</td>
</tr>
<tr>
<td>6.</td>
<td>The system, as much as possible should provide informative content to user.</td>
</tr>
<tr>
<td>7.</td>
<td>The system should allow the input of data to be done rapidly with the using of keyboard by following the sequence of field on the screen</td>
</tr>
<tr>
<td>8.</td>
<td>The values for Weight should ranging from 1 to 10</td>
</tr>
<tr>
<td>9.</td>
<td>The system will verify the input data before storing the data in the database by triggering an alert to the user if the value is invalid.</td>
</tr>
</tbody>
</table>

**5.3.8 Maintain Feature Module**

Main Feature modules allows user to input the features which will then linked to main contributor.
Figure 5.8: Use Case Diagram for Maintain Feature

Ten requirements must be fulfilled to implement the Maintain Feature module, as described in Table 5.8 below:

Table 5.8: The Requirement for the Feature

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a feature interface that consists of contributor code, feature code, feature name and numeric value.</td>
<td>CRM-Mod08-01</td>
</tr>
<tr>
<td>2.</td>
<td>The system shall allow user to sort the feature listing based on their preference.</td>
<td>CRM-Mod08-02</td>
</tr>
<tr>
<td>3.</td>
<td>The system should only allow numeric values in Numeric Values field</td>
<td>CRM-Mod08-03</td>
</tr>
<tr>
<td>4.</td>
<td>The system should allow user to use the same mode (new, save and delete) which is available in the other interfaces.</td>
<td>CRM-Mod08-04</td>
</tr>
</tbody>
</table>
5. The system should prompt users for confirmation each time user wanted to perform the task

6. The system, as much as possible should provide informative content to user.

7. The system should allow the input of data to be done rapidly with the using of keyboard by following the sequence of field on the screen

8. The values for Numeric Values should ranging from 1 to 20

9. The system should tied the contributor to feature list

10. The system will verify the input data before to storing the data in the database by triggering an alert to the user if the value is invalid.

---

**5.3.9 Change Password Module**

The Change Password modules allows user to change the Oracle password for security purpose. The passwords are strings of characters used to authenticate the users. If the user manages to login successfully, the system will trust that they are valid user and grant them access to the data.
Four requirements must be fulfilled to implement the Schedule Factory Plan module, as described in Table 5.9 below.

**Table 5.9: The Requirement for the Change Password**

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a Change Password interface that consists of old password and new password.</td>
<td>CRM-Mod09-01</td>
</tr>
<tr>
<td>2.</td>
<td>The system will make sure the old password is the current</td>
<td>CRM-Mod09-02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>The system will make sure the old password and current password values matched.</td>
<td>CRM-Mod09-03</td>
</tr>
<tr>
<td>4.</td>
<td>The new passwords must be a minimum of six characters with at least one numeric character and one alpha character.</td>
<td>CRM-Mod09-04</td>
</tr>
</tbody>
</table>
5.4 CONCLUSION

All the requirements of the system had been documented in this chapter. These requirements are derived from the analysis and review done in Literature Review (chapter 2) and Critical Analysis (chapter 3) which is done through the application of the use-case scenarios.
CHAPTER 6

SYSTEM DESIGN

6.0 SYSTEM ARCHITECTURE

According to the Rational Unified Process:

"Software architecture encompasses the significant decisions about the organization of a software system. The selection of the structural elements and their interfaces by which the system is composed together with their behavior as specified in the collaboration among those elements. The composition of the structural and behavioral elements into progressively larger subsystems, the architectural style that guides this organization, these elements, and their interfaces, their collaborations, and their composition. Software architecture is concerned not only with structure and behavior but also with usage, functionality, performance, resilience, reuse, comprehensibility, economic and technology constraints and trade-offs, and aesthetic issues."

Client server architecture had been chosen for this project. By using the relational database management system (DBMS), user queries could be answered directly. Besides that, it also reduced network traffic by providing a query response rather than total file transfer. It improves multi-user updating through a GUI front end to a shared database. In client/server architectures standard query language (SQL) statements are typically used to communicate between the client and server (Schussel, 1996).
The two-tier software architecture had been opted for this project. The two tier architecture is intended to improve usability by supporting a forms-based and user-friendly interface. It also improves scalability by accommodating up to 100 users and improves flexibility by allowing data to be shared. The two tier architecture requires minimal operator intervention, and is frequently used in non-complex, non-time critical information processing systems.

The two tier architectures consist of three components distributed in two layers: client (requester of services) and server (provider of services). The three components are:

1. User System Interface (such as session, text input, dialog, and display management services)

2. Processing Management (such as process development, process enactment, process monitoring, and process resource services)

3. Database Management (such as data and file services)

The two tier design allocates the user system interface exclusively to the client. It places database management on the server and splits the processing management between client and server, creating two layers. The following figure depicts the two tier software architecture.
6.1 CBR DESIGN

The system consists of four major modules which are:

i. Problem input

Problem input is the module where the marketing planner defines the contributor that contributes to the decision making process. It consists of stocks, demands, tyre size popularity, past month sales and factory capacity.

ii. System engine

The system engine consists of the following:
a. Case base – it is a database where all the cases are stored

b. CBR inference engine – it is a computer program that derives the answers from the knowledge base. It is the brain that the expert system uses to reason regarding the information in the knowledge base.

c. CBR knowledgebase – it is the representation of the cases. The nearest neighbor is implemented in the knowledge base and it had been explained in details in **Chapter 2 - Literature Review**.

The indexing rules are assigned to each input

a. Indexing Rules

   It consists of assigning the indices ID for each contributor code, feature code and case id, assigning the weight to the contributor code.

b. Similarity Rules

   By using the nearest neighbour algorithm for the similarity module, the target case global similarity value will be compared with other global similarity cases in the case base. The matching algorithm is based on the value of global similarity. Local similarity is the total value of multiplication of the weight and the differences between target case and the existed cases in the case base for particular feature. The global similarity is the total of the entire local similarity
feature. The equations for local similarity and the global similarity are as the following:

\[
\text{Local Similarity} = \sum_{i=1}^{n} W_i \times \text{sim} (f_i^I, f_i^R)
\]

\(W_i\) = the weight of the particular feature and \(\text{sim}\) is the similarity function,

\(f_i^I\) and \(f_i^R\) = the values for feature \(i\) in the input and retrieved cases respectively.

\[
\text{Global Similarity} = \text{Sum Total for all local similarity.}
\]

Later the global similarity is compared with the other 3 biggest global similarities that exist in the case base. If the global similarity value is greater then the biggest global similarity value, the new global similarity becomes as the new biggest global similarity.

If user input the problem descriptor and the target case global similarity value does not exist in the case base, then he needs to update it.

c. Adaptation Rules

The adaptation rule is to adapt a case to solve user problem. The system should gained experience every time when it is used to provide solution to present case. The system is capable to use the solution of previous case to solve the problem of present case where the previous case is similar to the present case. If the provided solution to the present case is suitable, the system should add this new case to the case base. With the increase cases of case base, the system will be
able to provide more accurate cases to user. A case consists of two parts which are the problem and the solution. For a new case that will be added to the case base by the user, the first part of new case is copy from the present case and the second part is copy from the matched case. The new case is the combination of previous case and present case.

d. Repair rule

The marketing planner will review the case in the case base to ensure the solution provided is right. If the solution is not correct and not satisfied, then he will need to amend the solution. The amendment version later is saved in the case base which would then overwrite the old solution.

iii. Output

The solution/decision of the moderation is shown to the user.
6.2 SYSTEM DOMAIN KNOWLEDGE

The main domain knowledge of the system is the moderation decision making. This is where the decision needs to be made whether the forecast figures submitted by the sales office need to be increase, maintain or reduced. The contributor, which had been keyed in by the user contributes to the solution of the moderation process. The input from the user consists of the stock, demands, past month sales, factory capacity and popularity.

6.3 USE CASE DESCRIPTION

The use case diagram is created to illustrate the relationships among use cases and actors and also the related use cases. It is the visual representation of what the user wants the system to do and represents a major piece of functionality that is complete from beginning to the end. The use case diagrams for this project had been illustrated in Chapter 5 - Requirement Analysis.

Actor is someone or something outside the system that must interact with the system, either by giving or receiving information or both. It is represented by a stickman in UML.

6.3.1 Sign up Module

The actor of this sign up module is the system administrator who is a powerful user to create the id for all the users in the system.
### Table 6.1: Sign up Use Case Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Sign Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>System administrator</td>
</tr>
<tr>
<td>Description</td>
<td>Register’s personal details</td>
</tr>
</tbody>
</table>

#### Course of Events

<table>
<thead>
<tr>
<th>Actor Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The actor key in employee code</td>
<td>2. The system display the employee name</td>
</tr>
<tr>
<td>3. The actor key in the network id</td>
<td>4. The user id and password will follow the network id</td>
</tr>
<tr>
<td>5. The actor saved the record</td>
<td>6. The system saves the transaction.</td>
</tr>
</tbody>
</table>

### 6.3.2 Access Authorization

Sales representatives, marketing planner and factory planner are the actors who are allowed to access the Access Authorization module.

### Table 6.2: Access Authorization Use Case Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Access Authorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Sales office representative, Marketing planner, Factory planner</td>
</tr>
<tr>
<td>Description</td>
<td>Log in inside the system</td>
</tr>
</tbody>
</table>

#### Course of Events

<table>
<thead>
<tr>
<th>Actor Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The actor key in username</td>
<td>5. The system verifies the parameters keyed-in.</td>
</tr>
<tr>
<td>2. The actor key in password</td>
<td>6. If connection to the database</td>
</tr>
<tr>
<td>3. The actor press tab at Database name field</td>
<td></td>
</tr>
<tr>
<td>4. The actor clicks on Login button</td>
<td></td>
</tr>
</tbody>
</table>
6.3.3 Consolidate Forecast Module

In this module, only marketing planner is allowed to access the consolidate forecast module. This is because only the marketing planner is allowed to consolidate the data from all the sales office and product group.

<table>
<thead>
<tr>
<th>Name</th>
<th>Consolidate forecast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Marketing planner</td>
</tr>
<tr>
<td>Description</td>
<td>Consolidate the forecast</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course of Events</th>
<th>Actor Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. The actor choose the product group</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. The actor choose the transaction date</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. The actor choose the sales office</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. The actor click on View button</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. The system display the forecast details submitted by the sales office</td>
<td></td>
</tr>
</tbody>
</table>
### 6.3.4 Populate Graph Module

**Table 6.4: Populate Graph Use Case Description**

<table>
<thead>
<tr>
<th>Name</th>
<th>Populate graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Marketing planner</td>
</tr>
<tr>
<td>Description</td>
<td>Generate the graph</td>
</tr>
</tbody>
</table>

#### Course of Events

<table>
<thead>
<tr>
<th>Actor Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The actor choose the product group.</td>
<td>7. The system populates the graph based on the popular items.</td>
</tr>
<tr>
<td>2. The actor choose the transaction date.</td>
<td></td>
</tr>
<tr>
<td>3. The actor choose the type of graph to be display.</td>
<td></td>
</tr>
<tr>
<td>4. The actor key in the forecast month.</td>
<td></td>
</tr>
<tr>
<td>5. The actor key in the quota values to be display on the graph.</td>
<td></td>
</tr>
<tr>
<td>6. The actor click on View button.</td>
<td></td>
</tr>
</tbody>
</table>
### 6.3.5 Moderate Forecast Module

**Table 6.5: Moderate Forecast Use Case Description**

<table>
<thead>
<tr>
<th>Name</th>
<th>Moderate forecast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Marketing planner</td>
</tr>
<tr>
<td>Description</td>
<td>Amend the forecast</td>
</tr>
<tr>
<td>Course of Events</td>
<td></td>
</tr>
<tr>
<td><strong>Actor Action</strong></td>
<td></td>
</tr>
<tr>
<td>1. The actor choose the product group</td>
<td></td>
</tr>
<tr>
<td>2. The actor choose the transaction date</td>
<td></td>
</tr>
<tr>
<td>3. The actor click on View button</td>
<td></td>
</tr>
<tr>
<td>5. The actor change the forecast values</td>
<td></td>
</tr>
<tr>
<td>6. The actor click on Save button</td>
<td><strong>System Response</strong></td>
</tr>
<tr>
<td>4. The system displays the forecast values</td>
<td></td>
</tr>
<tr>
<td>7. The system saves the transactions</td>
<td></td>
</tr>
</tbody>
</table>

### 6.3.6 Schedule Factory Plan Module

**Table 6.6: Schedule Factory Plan Use Case Description**

<table>
<thead>
<tr>
<th>Name</th>
<th>Schedule factory plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Factory planner</td>
</tr>
<tr>
<td>Description</td>
<td>Schedule the factory plan</td>
</tr>
<tr>
<td>Course of Events</td>
<td></td>
</tr>
<tr>
<td><strong>Actor Action</strong></td>
<td></td>
</tr>
<tr>
<td>1. The actor choose the product group</td>
<td></td>
</tr>
<tr>
<td>2. The actor choose the transaction date</td>
<td></td>
</tr>
<tr>
<td>3. The actor click on View button</td>
<td><strong>System Response</strong></td>
</tr>
<tr>
<td>4. The system displays the moderated forecast on the screen</td>
<td></td>
</tr>
</tbody>
</table>
5. The actor schedule the factory plan and click on Save button

6. The system saves the transaction

### 6.3.7 Maintain Main Contributor Module

**Table 6.7: Maintain Main Contributor Use Case Description**

<table>
<thead>
<tr>
<th>Name</th>
<th>Maintain Contributor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Marketing planner</td>
<td>Maintain the main contributor inside the system</td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course of Events</th>
<th>Actor Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. The actor key in the main contributor code</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. The actor key in the contributor name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. The actor key in the description</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. The actor click on Save button</td>
<td>5. The system saves the transaction</td>
</tr>
<tr>
<td></td>
<td>to save the record</td>
<td></td>
</tr>
</tbody>
</table>

### 6.3.8 Maintain Feature Module

**Table 6.8: Maintain Feature Use Case Description**

<table>
<thead>
<tr>
<th>Name</th>
<th>Maintain Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Marketing planner</td>
<td>Maintain feature inside the system</td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course of Events</th>
<th>Actor Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. The actor chooses main contributor from the drop down list</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. The actor key in feature code</td>
<td></td>
</tr>
</tbody>
</table>
3. The actor key in feature name
4. The actor key in description
5. The actor saves the record

6. The system saves the transaction

6.3.9 Change Password Module

Table 6.9: Change Password Use Case Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Change Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Sales office representative, Marketing planner, Factory planner</td>
</tr>
<tr>
<td>Description</td>
<td>Change the password of the system</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course of Events</th>
<th>Actor Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The actor key in old password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. The actor key in new password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. The actor re-key in new password to confirm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. The actor click OK button</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. The system checks if the new password matches the reconfirm password. If matched, the system prompts the password had been changed successfully.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 6.3.10 Maintain the User Case Module

**Table 6.10: Maintain the User Use Case Description**

<table>
<thead>
<tr>
<th>Name</th>
<th>Maintain the User</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>Marketing planner</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Maintain the user use case</td>
<td></td>
</tr>
<tr>
<td>Course of Events</td>
<td><strong>Actor Action</strong></td>
<td><strong>System Response</strong></td>
</tr>
<tr>
<td></td>
<td>1. The actor chooses the transaction date</td>
<td>2. The system displays the use case for the selected month (if any) and the tyre size available in the forecast</td>
</tr>
<tr>
<td></td>
<td>3. The actor chooses the item</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. The actor chooses the Stock</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. The actor chooses the Past Month Sales</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6. The actor chooses the Demand</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7. The actor chooses the Factory Capacity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8. The actor chooses the Popularity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9. The actor clicks on Check button</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10. The system checks whether there is similarity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11. System prompts message if found the similarity or else system prompt no record found</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12. The actor chooses the moderation decision</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13. The actor click on Save button</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14. The system saves the transaction</td>
<td></td>
</tr>
</tbody>
</table>
6.4 ACTIVITY DIAGRAM

Activity diagram is used in process modeling and analysis of the requirement stages of UML. It is used for documenting existing processes, analyzing new process concept, identifying IT levers and finding reengineering opportunities. It is used to show parallel behavior between different events and activities. All the activity diagrams are available in Appendix F – Activity Diagram.

6.5 SEQUENCE DIAGRAM

Sequence diagram is an interaction diagram which emphasizes the time ordering of messages. It is the primary source of information used to determine class responsibilities and interfaces. All the sequence diagrams are shown in the following diagrams:
Figure 6.2: Login Sequence Diagram

Figure 6.3: Maintain Contributor Sequence Diagram
Figure 6.4: Maintain Feature Sequence Diagram

Figure 6.5: Maintain the User Sequence Diagram
Figure 6.6: Consolidate Forecast Sequence Diagram

Figure 6.7: Populate Graph Sequence Diagram
Figure 6.8: Moderate Forecast Sequence Diagram

Figure 6.9: Schedule Factory Plan Sequence Diagram
6.6 SCREEN DESIGN

The design of the screens had gone through three iteration cycle in order for user to feel confidence that it will meet their user requirement. The first iteration is done based on paper prototype in which it was done using sketching (refer Appendix G-Paper Prototype Sketches, Figure F1 - F9)

6.6.1 The First Iteration

Based on the feedback given by the user (refer Table 6.11), some amendment had been done to the sketches.

Table 6.11: Feedback from First Iteration of Prototyping

<table>
<thead>
<tr>
<th>Screens</th>
<th>Evaluation from user</th>
</tr>
</thead>
</table>
| Login screen          | 1. OK button should be change to Login button and Cancel button should be change to Exit  
                          2. The selection of database name should be part of Login parameters and it should be defaulted to current database |
| Company selection screen | 1. Only company granted to the user should be listed in the screen                        |
| Main contributor screen | None                                                                                   |
| Feature screen       | None                                                                                   |
| Forecast screen       | 1. The forecast, visualization and moderation screen should be combine and put inside one screen so that it will be easier to navigate  
                          2. Date should be change to Month                                                      |
| Visualization screen  | 1. The month of the forecast should be made as parameter driven for user to key in  
                          2. The items should be sorted based on popularity order                                 |
3. Date should be changed to Month

<table>
<thead>
<tr>
<th>Screens</th>
<th>Evaluation from user</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moderation screen</td>
<td>1. Date should be changed to Month</td>
</tr>
<tr>
<td>Factory plan screen</td>
<td>1. The details on the grid should be separated by section</td>
</tr>
</tbody>
</table>

### 6.6.2 The Second Iteration

Upon receiving feedback from users in the first iteration, all the changes had been included inside the system. Visual Basic 6 had been using as a tool to come out with the prototype in the second iteration (refer Appendix H-System Prototype, Figure H1 to H8).

<table>
<thead>
<tr>
<th>Screens</th>
<th>Evaluation from user</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login screen</td>
<td>1. The status bar should be included to show the progress of the database connectivity</td>
</tr>
<tr>
<td></td>
<td>2. The copyright should be incorporated inside the screen</td>
</tr>
<tr>
<td>Company selection screen</td>
<td>1. If the user press Esc, by default the first company will be defaulted as company chosen</td>
</tr>
<tr>
<td>Main contributor screen</td>
<td>None</td>
</tr>
<tr>
<td>Feature screen</td>
<td>None</td>
</tr>
<tr>
<td>Moderation screen</td>
<td>Forecast</td>
</tr>
<tr>
<td></td>
<td>-None</td>
</tr>
<tr>
<td></td>
<td>Visualization</td>
</tr>
<tr>
<td></td>
<td>1. The sorting of the items on the graph should be based on the popularity of the items</td>
</tr>
<tr>
<td>Marketing Moderation</td>
<td>1. The popular items should be highlighted so that it would be easier for user to notice them</td>
</tr>
<tr>
<td></td>
<td>2. The number of populate items should be make as parameter driven</td>
</tr>
<tr>
<td>Factory plan screen</td>
<td>1. Subtotal of the following items should be included:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>
|   | a. Moderated quantity  
b. Accepted program  
2. Remarks should be included in accepted program so that production planner will be able to input reason if he cannot meet the quantity requested by marketing (for recording purpose) |
6.7 DATABASE DESIGN

Database design is the process of producing a detailed data model of a database. Following are the table structure with each field’s explanation. The primary key that enforce the entity integrity by uniquely identifying entity instances is created for each table.

1. **Table name**: CRM_USERS_M
   
   **Primary key**: CRM_USERS_PK (userid)
   
   **Table description**: The tables stores the user’s personal details.
   
   **Table 6.13: User master table structure**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Companycode</td>
<td>Varchar2(8)</td>
<td>Not null</td>
<td>Company id</td>
</tr>
<tr>
<td>Employeecode</td>
<td>Varchar2(8)</td>
<td>Not null</td>
<td>Employee id</td>
</tr>
<tr>
<td>Employeename</td>
<td>Varchar2(50)</td>
<td>Not null</td>
<td>Employee name</td>
</tr>
<tr>
<td>Userid</td>
<td>Varchar2(30)</td>
<td>Not null</td>
<td>User id</td>
</tr>
<tr>
<td>Network id</td>
<td>Varchar2(30)</td>
<td>Not null</td>
<td>Network id</td>
</tr>
</tbody>
</table>

2. **Table name**: CRM_CONTRIBUTOR_M
   
   **Primary key**: CRM_CONTRIBUTOR_PK (contributorcode)
   
   **Table description**: The table stores the contributor details with weight associated with the contributor.
   
   **Table 6.14: Contributor master table structure**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributorcode</td>
<td>Varchar2(5)</td>
<td>Not null</td>
<td>Contributor id</td>
</tr>
<tr>
<td>Contributorname</td>
<td>Varchar2(50)</td>
<td></td>
<td>Contributor description</td>
</tr>
<tr>
<td>Weight</td>
<td>Number(3)</td>
<td></td>
<td>Weight</td>
</tr>
</tbody>
</table>
3. **Table name: CRM_FEATURE_M**

Primary key: CRM_FEATURE_PK(contributorcode, featurecode)

Table description: The table stores the feature, contributor code which the feature is group and numeric value for each feature.

**Table 6.15: Feature master table structure**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributorcode</td>
<td>Varchar2(5)</td>
<td>Not null</td>
<td>Contributor code</td>
</tr>
<tr>
<td>Featurecode</td>
<td>Varchar2(5)</td>
<td>Not null</td>
<td>Feature id</td>
</tr>
<tr>
<td>Featurename</td>
<td>Varchar2(50)</td>
<td></td>
<td>Feature name</td>
</tr>
<tr>
<td>Numvalue</td>
<td>Number(2)</td>
<td></td>
<td>Numeric value</td>
</tr>
</tbody>
</table>

4. **Table name: CRM_FSE_M**

Table description: The table store the sales forecast submitted by each sales office. The forecast is based on item code, which is stored on monthly basis and kept for four consecutive months.

**Table 6.16: Forecast transaction table structure**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Companycode</td>
<td>Varchar2(8)</td>
<td>Not null</td>
<td>Company code of the sales office</td>
</tr>
<tr>
<td>Salesofficecode</td>
<td>Varchar2(10)</td>
<td>Not null</td>
<td>Sales office/branches that submits the forecast</td>
</tr>
<tr>
<td>Warehouse_id</td>
<td>Varchar2(10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fsemonth</td>
<td>Varchar2(2)</td>
<td></td>
<td>Month of the transaction, example 02 for February month</td>
</tr>
<tr>
<td>Fseyear</td>
<td>Varchar2(4)</td>
<td></td>
<td>Year of the transaction, example 2006 for year 2006</td>
</tr>
<tr>
<td>Itemcode</td>
<td>Varchar2(10)</td>
<td></td>
<td>Item code/size code of tyre</td>
</tr>
<tr>
<td>Fse1</td>
<td>Number(9)</td>
<td></td>
<td>Month 1 of the forecast</td>
</tr>
<tr>
<td>Fse2</td>
<td>Number(9)</td>
<td></td>
<td>Month 2 of the forecast</td>
</tr>
<tr>
<td>Fse3</td>
<td>Number(9)</td>
<td></td>
<td>Month 3 of the forecast</td>
</tr>
<tr>
<td>Fse4</td>
<td>Number(9)</td>
<td></td>
<td>Month 4 of the forecast</td>
</tr>
<tr>
<td>Fse5</td>
<td>Number(9)</td>
<td></td>
<td>Month 5 of the forecast</td>
</tr>
<tr>
<td>Fse6</td>
<td>Number(9)</td>
<td></td>
<td>Month 6 of the forecast</td>
</tr>
<tr>
<td>Fse7</td>
<td>Number(9)</td>
<td></td>
<td>Month 7 of the forecast</td>
</tr>
<tr>
<td>Fse8</td>
<td>Number(9)</td>
<td></td>
<td>Month 8 of the forecast</td>
</tr>
<tr>
<td>Field</td>
<td>Data type, Null?</td>
<td>Remarks</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Fse9</td>
<td>Number(9),</td>
<td>Month 9 of the forecast</td>
<td></td>
</tr>
<tr>
<td>Fse10</td>
<td>Number(9),</td>
<td>Month 10 of the forecast</td>
<td></td>
</tr>
<tr>
<td>Fse11</td>
<td>Number(9),</td>
<td>Month 11 of the forecast</td>
<td></td>
</tr>
<tr>
<td>Fse12</td>
<td>Number(9),</td>
<td>Month 12 of the forecast</td>
<td></td>
</tr>
<tr>
<td>Onhandqty</td>
<td>Number(9),</td>
<td>Quantity of on hand stock</td>
<td></td>
</tr>
<tr>
<td>Availstock</td>
<td>Number(9),</td>
<td>Quantity of available stock</td>
<td></td>
</tr>
</tbody>
</table>

5. Table name: CRM_MARKET_M

Primary key: CRM_MARKET_PK (companycode, marketcode)

Table description: The table stores the market master for each company.

**Table 6.17: Market master table structure**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Companycode</td>
<td>Varchar2(8)</td>
<td>Not null</td>
<td>Company code of the sales office</td>
</tr>
<tr>
<td>Marketcode</td>
<td>Varchar2(10)</td>
<td>Not null</td>
<td>Market code</td>
</tr>
<tr>
<td>Marketname</td>
<td>Varchar2(50)</td>
<td></td>
<td>Market description</td>
</tr>
<tr>
<td>Fse</td>
<td>Number(1)</td>
<td></td>
<td>Market indicator for forecast, which indicates market available for forecast</td>
</tr>
</tbody>
</table>

6. Table name: CRM_MAPMARKET_M

Table description: The tables stores the mapping between sales office and market.

**Table 6.18: Market mapping master table structure**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salesofficecode</td>
<td>Varchar2(10)</td>
<td></td>
<td>Sales office/branches code</td>
</tr>
<tr>
<td>Marketcode</td>
<td>Varchar2(10)</td>
<td></td>
<td>Market code</td>
</tr>
</tbody>
</table>

7. Table name: CRM_MODERATE_M

Primary key: CRM_MODERATE_PK (moderatecode)

Table description: The table stores the moderation decision list.
Table 6.19: Moderation master table structure

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moderatecode</td>
<td>Varchar2(8)</td>
<td>Not null</td>
<td></td>
</tr>
<tr>
<td>Moderatename</td>
<td>Varchar2(1000)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. Table name: CRM_PRODUCTGRP_M

Primary key: CRM_PRODUCTGRP_PK (productgrpcode)

Table description: The tables stores the product group master.

Table 6.20: Product group master table structure

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Productgrpcode</td>
<td>Varchar2(8)</td>
<td>Not null</td>
<td>Product group code of tyre</td>
</tr>
<tr>
<td>Productgrpname</td>
<td>Varchar2(50)</td>
<td></td>
<td>Product group name of tyre</td>
</tr>
<tr>
<td>Description</td>
<td>Varchar2(100)</td>
<td></td>
<td>Remarks/description</td>
</tr>
</tbody>
</table>

9. Table name: CRM_SALESOFFICE_M

Primary key: CRM_SALESOFFICE_PK (companycode, salesofficecode)

Table description: The table stores the sales office master for each company.

Table 6.21: Sales office master table structure

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Companycode</td>
<td>Varchar2(8)</td>
<td>Not null</td>
<td>Company code of the sales office</td>
</tr>
<tr>
<td>Salesofficecode</td>
<td>Varchar2(10)</td>
<td>Not null</td>
<td>Sales office/branches code</td>
</tr>
<tr>
<td>Salesofficename</td>
<td>Varchar2(50)</td>
<td></td>
<td>Sales office/branches name</td>
</tr>
</tbody>
</table>

10. Table name: CRM_MARKETMODERATE_T

Table description: The table stores the forecast figures that had been moderated based on each item and market.

Table 6.22: Market moderation transaction table structure

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Companycode</td>
<td>Varchar2(8)</td>
<td></td>
<td>Company code</td>
</tr>
</tbody>
</table>
11. Table name: CRM_ACCEPTPLAN_T

Primary key: CRM_ACCEPTPLAN_PK (companycode, plnmonth, plnyear, itemcode)

Table description: The table stores the information on the planning of the tyre production on weekly basis.

**Table 6.23: Accept plan transaction table structure**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Companycode</td>
<td>Varchar2(8)</td>
<td>Not null</td>
<td>Company code</td>
</tr>
<tr>
<td>Plnmonth</td>
<td>Varchar2(2)</td>
<td>Not null</td>
<td>Planning month, example 02 for February month</td>
</tr>
<tr>
<td>Plnyear</td>
<td>Varchar2(4)</td>
<td>Not null</td>
<td>Planning year, example 2006 for year 2006</td>
</tr>
<tr>
<td>Itemcode</td>
<td>Varchar2(10)</td>
<td>Not null</td>
<td>Item code/size code</td>
</tr>
<tr>
<td>Week1</td>
<td>Number(5,2)</td>
<td></td>
<td>Week 1 of the month</td>
</tr>
<tr>
<td>Week2</td>
<td>Number(5,2)</td>
<td></td>
<td>Week 2 of the month</td>
</tr>
<tr>
<td>Week3</td>
<td>Number(5,2)</td>
<td></td>
<td>Week 3 of the month</td>
</tr>
<tr>
<td>Week4</td>
<td>Number(5,2)</td>
<td></td>
<td>Week 4 of the month</td>
</tr>
<tr>
<td>Week5</td>
<td>Number(5,2)</td>
<td></td>
<td>Week 5 of the month</td>
</tr>
<tr>
<td>Remarks</td>
<td>Varchar2(1000)</td>
<td></td>
<td>Remarks/comments</td>
</tr>
</tbody>
</table>

12. Table name: CRM_CASEDETAIL_T

Table description: The table stores the input for each feature and the value associated with it.

**Table 6.24: Case details transaction table structure**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Data type</th>
<th>Null?</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casemonth</td>
<td>Varchar2(2)</td>
<td></td>
<td>Month</td>
</tr>
<tr>
<td>Caseyear</td>
<td>Varchar2(4)</td>
<td></td>
<td>Year</td>
</tr>
<tr>
<td>Casedetcode</td>
<td>Varchar2(8)</td>
<td></td>
<td>Case detail Id code</td>
</tr>
<tr>
<td>Field name</td>
<td>Data type</td>
<td>Null?</td>
<td>Remarks</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>-------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Casemonth</td>
<td>Varchar2(2)</td>
<td></td>
<td>Month</td>
</tr>
<tr>
<td>Caseyear</td>
<td>Varchar2(4)</td>
<td></td>
<td>Year</td>
</tr>
<tr>
<td>Casecode</td>
<td>Varchar2(8)</td>
<td></td>
<td>Case code</td>
</tr>
<tr>
<td>Totvalue</td>
<td>Number(7,2)</td>
<td></td>
<td>Total value</td>
</tr>
<tr>
<td>Moderatecode</td>
<td>Varchar2(8)</td>
<td></td>
<td>Moderation code</td>
</tr>
</tbody>
</table>
6.8 CONCLUSION

The overview of the design that is discussed in this chapter was based on the requirements elicited in *Chapter 5-Requirement Analysis*. The relationship between components, modules, interfaces and data for the system had been shown in the chapter in order to satisfy the requirements elicited in *Chapter 5-Requirement Analysis*. 
CHAPTER 7

IMPLEMENTATION

7.0 INTRODUCTION

This chapter will explain the system environment employed during the development of the system. The verification of each requirement and system design follows after each of the module interfaces is shown.

7.1 DESIGN PRINCIPLES

Design principle is another way of conceptualizing usability. It is derived from a combination of theory-based knowledge, experience and common sense. It tends to be written in a prescription manner, suggesting to developers what to provide and what to avoid at the interface, such as the Do and Don’ts of the interaction design. Among the common design principles or rules that had been applied in this project are as the following:

7.1.1 Consistency

Consistency refers to designing interfaces to have similar operations and similar elements in order to achieve tasks. In particular, a consistent interface is the one that follows rules, such as using the same operation to print reports. Inconsistent interface, on the other hand allow exceptions to a rule. Problem with the inconsistency is that it is quite
arbitrary, that make it difficult for users to remember and make users more prone to errors.

In the system, the consistency is applied throughout the entire screen in a sense that all the screens are using the standard toolbar to perform the same operation.

Regardless of any screens, when a user clicks the New button on the toolbar, then all the values on the screen will be cleared and the system is ready to receive any input from the user. This is supported in the following codes:

```
01 Public Sub Addrecord()
02    Toolbar.Buttons("Delete").Enabled = False
03    txtFeatureCode.Text = ""
04    txtFeatureName.Text = ""
05    cboContributorCode.Text = ""
06    txtValue.Text = 0
07    cboContributorCode.SetFocus
08 End Sub
```

Add record procedure in Feature screen

```
09 Public Sub Addrecord()
10    Toolbar.Buttons("Delete").Enabled = False
11    txtContributorCode.Text = ""
12    txtContributorDesc.Text = ""
13    txtWEight.Text = 0
14    txtContributorCode.SetFocus
15 End Sub
```

Add record procedure in Contributor screen

**Line 03-06 and Line 11-13** shows that all the existing values will be cleared.
Line 07 and Line 14 shows that the cursor is placed to the first field on the screen once all the values are cleared, to show that the system is ready for data input.

When user click on Save button on the toolbar in any screens, system will prompt an error message asking whether user wants to continue with the operation. Then the records will be saved upon confirmation. A similar confirmation message box is prompted to the user in all the screens upon saving the records.

\[\text{iResp} = \text{MsgBox("Are you sure you want to save the record,"} \text{& txtFeatureCode.Text & \\
\text{"-"} \text{& txtFeatureName.Text & \\
\text{"?"}, vbExclamation + vbYesNoCancel)}\]

Saving message box in Feature screen

\[\text{iResp} = \text{MsgBox("Are you sure you want to save the record,"} \text{& txtContributorCode} \text{& \\
\text{"-"} \text{& txtContributorDesc.Text & \\
\text{"?"}, vbExclamation + vbYesNoCancel)}\]

Saving message box in Contributor screen

The same scenario takes place when user clicks on Delete button. The selected record will be deleted upon confirmation as shown in the following codes:

\[\text{Res} = \text{MsgBox("Do you want to delete this record,"} \text{& txtFeatureCode.Text & \\
\text{"-"} \text{& txtFeatureName.Text & \\
\text{"?"}, vbQuestion + vbYesNo)}\]

Deleting message box in Contributor screen

\[\text{Res} = \text{MsgBox("Do you want to delete this record,"} \text{& txtContributorCode.Text & \\
\text{"-"} \text{& txtContributorDesc.Text & \\
\text{"?"}, vbQuestion + vbYesNo)}\]

Deleting message box in Contributor screen

This can be explained from the following codes:

01 Private Sub Toolbar_ButtonClick(ByVal Button As MsoControlLib.Button)
02   On Error GoTo Error_Handle
03   Select Case Button.key
04     Case "New”
Screen.ActiveForm.Addrecord

Case “Find”
Screen.ActiveForm.SearchRecord

Case “Save”
Screen.ActiveForm.saveRecord

Case “Delete”
Screen.ActiveForm.deleteRecord

Case “Info”
Screen.ActiveForm.ShowInfo

Case “Cancel”
Screen.ActiveForm.CancelRecord

Case “Close All”
If Not (ActiveForm Is Nothing) Then
    If Toolbar.Buttons(“Save”).Enabled Then
        If MsgBox(“Do you want to save your changes?”, vbYesNo + vbQuestion, “Close Form”) = vbYes Then
            ActiveForm.Document.saveRecord
        End If
    End If
End If

Dim msg$, tmp As Integer

msg$ = “All open forms will be closed.” & vbCrLf & vbCrLf
msg$ = msg & “Do you wish to proceed?”

Beep

tmp = MsgBox(msg$, vbYesNo + vbQuestion, “Close All Forms”)

If tmp = vbNo Then Exit Sub

MousePointer = vbHourglass
CloseAccept = False
ShowMsg “Please wait for the form(s) to close...”
Do Until (ActiveForm Is Nothing)
    Unload ActiveForm
    If CloseAccept = True Then Exit Do
Loop

TriggerToolbar

MousePointer = vbDefault
ShowMsg "Ready"
End If

Case "Print"
Screen ActiveForm.PrintRecord

Case "Help"
mnuSubHelp_Click

Case "Exit"
mnuExit_Click
End Select

Exit Sub

Error_Handle:
MsgBox Err.Number & " – " & Err.Description, vbCritical, "System Error"

Exit Sub
End Sub

The above coding is extracted from the Main menu. Each button in the toolbar has its own name like New, Find, Save, Delete, Info, Cancel, Close All, Print and Exit.

Table 7.1: Functionality of each buttons

<table>
<thead>
<tr>
<th>Button Name</th>
<th>Coding Line No</th>
<th>What does it do</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Line 05</td>
<td>Global call for adding new records</td>
</tr>
<tr>
<td>Find</td>
<td>Line 07</td>
<td>Global call for search records</td>
</tr>
<tr>
<td>Save</td>
<td>Line 09</td>
<td>Global call for saving records</td>
</tr>
<tr>
<td>Delete</td>
<td>Line 11</td>
<td>Global call for deleting records</td>
</tr>
<tr>
<td>Info</td>
<td>Line 13</td>
<td>Global call for showing info</td>
</tr>
<tr>
<td>Cancel</td>
<td>Line 15</td>
<td>Global call for canceling records</td>
</tr>
<tr>
<td>Close All</td>
<td>Line 32</td>
<td>Global call for closing active forms</td>
</tr>
<tr>
<td>Print</td>
<td>Line 40</td>
<td>Global call for printing records</td>
</tr>
<tr>
<td>Exit</td>
<td>Line 44</td>
<td>Global call for exiting the system</td>
</tr>
</tbody>
</table>
7.1.2 Visibility

The more visible functions are, the more likely the users will be able to know what to do next. If there is lack of visibility, the functions are out of sight which makes the user difficult to know how to use the system.

![Icons in CRMS](image1)

**Figure 7.2: Icons in CRMS**

The picture associated with the operation in the toolbar is standard and is used in other application such as Microsoft Word and Microsoft Excel. Even without looking at the Word, by looking at the picture, users are able to know what kind of the operation it performs. The buttons for different operations are clearly visible in the system.

![Icons in Microsoft Word](image2)

**Figure 7.3: Icons in Microsoft Word**

Besides that, the Exit button on the tool bar had been positioned to the far left of the screen because it makes it easier for the user since the Close button any other application is also position at the very far left of the screen.

![Close Button](image3)

**Figure 7.4: Close Button**
7.1.3 Feedback

Feedback is about sending back information about what action had been performed and what had been accomplished, allowing the person to continue with the activity. In CRMS, each time the user click on Log Off menu, the system will prompt a message box, informing user that by logging off, all the open forms will be close. It will then asking user for confirmation whether they really want to log off. The user is given two options, Yes, which means they want to log off and agree that the system will close all the open forms. System will then prompts user to wait for the forms to be closed. If he/she chooses No, that means they do not want to log off and system will not close the open forms. By providing the feedback to user, then they are able to know what the implications of their actions are, and therefore able to know what they had performed.

![Figure 7.5: Log off Submenu](image)

```vbnet
01 Private Sub mnuLogOff_Click()
02   Dim msg$, tmp As Integer
03   If Forms.Count > 1 Then
04      msg$ = “All open forms will be closed.” & vbCrLf & vbCrLf
05      msg$ = msg$ & “Do you wish to proceed?”
06      Beep
07      tmp = MsgBox(msg$, vbYesNo + vbQuestion, “Log Off”)
08      If tmp = vbNo Then Exit Sub
09      MousePointer = vbHourglass
10      ShowMsg “Please wait for the form(s) to close...”
```
In Line 04, system is alerting user that by logging off, all the open forms will be closed. User is given an option whether they would want to continue is shown in Line 07. By logging off, a feedback is prompted to user, as shown in Line 10, whereby the system informs the user to wait while the open forms are closing.

### 7.1.4 Constraints

The design of constraining refers to the determining ways of restricting the kind of user interaction that can take place at a given moment. One of the common design practice is to deactivate the menu options by disabled them, depending on the user access. By dong this, will then reduce the chances of making mistakes. In CRMS, the menu options are controlled by the user access. If the users do not have an access to the screen, then the menu will be disabled and this will prevent them from making mistakes.

![Disabled menu in CRMS](Figure 7.6: Disabled menu in CRMS)
Besides that, the display of the company in the Company Selection screen is also based on user company access. Only the company which the user has an access, will be listed in the Company Selection screen.

![Companies access granted to system:](image)

**Figure 7.7: Company Access in Company Selection Screen**

This is supported by the following coding:

```sql
sql = "select distinct a.*, b.companyname "
sql = sql & " from crm_usercomp a, crm_company_m b "
sql = sql & " where upper(username) = '" & Ucase(UserID) & "' "
sql = sql & " and a.companycode = b.companycode(+)"
```

In the above codes, the company access is based on user. The company master list consists of more company but since the user is only having to two companies, therefore only two companies are shown for the selection.

Besides that the buttons on the toolbar is also disabled when certain operations are performed. For example when New button is click, the button Delete is disable. This is to avoid user from making mistakes by deleting the records when he/she is creating a new
record. This is supported through the following code in Addrecord module in Feature screen:

```
01 Public Sub Addrecord()
02   Toolbar.Buttons("Delete").Enabled = False
03   txtFeatureCode.Text = ""
04   txtFeatureName.Text = ""
05   cboContributorCode.Text = ""
06   txtValue.Text = 0
07   cboContributorCode.SetFocus
08 End Sub
```

**Line 02** shows that the button Delete on the toolbar is disabled when new record is being created.

```
09 Public Sub Saverecord()
10     MsgBox "Record(s) had been saved", vbInformation
11     Toolbar.Buttons("Delete").Enabled = True
12     Exit Sub
14 errHandler:
15     Debug.Print Err.Number & Err.Description
16     MsgBox Err.Number & Err.Description
17 End Sub
```

And the button is enable back (as shown in **Line 12**) when the record is successfully saved as shown in the save Record module.

### 7.1.5 Closure

Human beings always want to know when the task performed had been complete so that they would be able to perform another task in the next level. In CRMS, whenever user had completed performing any operations like Save or Delete, a message box will be prompted to tell them that the record(s) had been saved/deleted successfully.
Public Sub DeleteRecord()
On Error GoTo errHandler

Static i%
Dim Res As Integer

If Trim$(txtFeatureCode.Text) = "" Then Exit Sub

For I = 1 To Feature.Feature.Count
    Res = MsgBox("Do you want to delete this record," & 
txtFeatureCode.Text & "-" & txtFeatureName.Text & ",?", vbQuestion + 
vbYesNo)
    If Res <> vbYes Then Exit Sub
    Feature.DeleteData Trim$(txtFeatureCode.Text)
    If Not Feature.FeatureErr Then
        populateFeature
        Addrecord
        MsgBox "Record had been deleted", vbInformation
        End If
    Exit Sub
Next
MsgBox "This Feature Code does not exist", vbExclamation
txtFeatureCode.Text = ""

Exit Sub
errHandler:
Debug.Print Err.Number & Err.Description
MsgBox Err.Number & Err.Description
End Sub

It is supported in the delete module in Feature screen, in which upon the deletion of the record (Line 09), the system prompted user, informing them that the record has been deleted (Line 14).

7.1.6 Prevent Errors

Safety means protecting the user from making mistakes and dangerous condition. The program should be safer in preventing user from making serious errors by reducing
the risk of mistaken actions and also providing users with recovery should errors happened. In CRMS, each time the user click on Close All button, the system will prompt a message box, asking user for confirmation whether he/she really wants to close the form. The user is given two options, which is Yes which means he wants to close the form and system will close all the open forms. Is he/she chooses No, that means he/she does not want to close the form and system will not close the form. By providing the safe interactive system, it should endanger confidence and allow user to use the system safely.

![Figure 7.8: Close All Button](image)

![Figure 7.9: Confirmation Message on Closing Forms](image)
7.1.7 Reduce Short Term Memory Load

Research showed that the capacity of short term memory is 7 plus 0 minus 2 chunks of information (the details are explained in Chapter 2 – Literature Review). Therefore, these had been made available in the systems whereby the length of password is set not to be more than 9 characters since chunking is not allowed in setting up the password.

```
01 Private Sub cmdChange_Click()
02   Dim sql$ 
03   Dim Rec   As Object 
04   Dim intProfileNo As Integer 
05    If Len(txtNewPwd.Text) > 10 Then  
06       MsgBox “The length of password should not be more than 9 char”, vbCritical  
07       Exit Sub 
08  End If 
09  ..
```

The validation of the length of the new password is done in Change Password screen when users click on Save button. The validation is done from line no 05 to 08. Any new password that had been set more than 9 characters will trigger a message box to the user. The chunking of information is applied in all the figures in the system, in which the numbers are chunks using thousand separators. For example quantity forecast of 12356 had been chunk to 2 groups, separated by comma; 12, 356. By using chunking, user will find it easier to remember the number and therefore can allocate their quality time in learning other features in the system.
7.1.8 Provide Informative Content

The tool tip is a text displayed on the screen when user paused a mouse on the control over it. It will provide brief content on the description of the controls on the screen. In order to do so, the Tool Tip Text property for the control need to be set in the Properties box.

Figure 7.10: Tool Tip
This is set in the properties box for the control (in this case, it is a text box) in Visual Basic as shown in the following screen shot:

![Figure 7.11: Setting up Tool Tip in Property Box](image)

Figure 7.11: Setting up Tool Tip in Property Box
7.2 USABILITY GOALS

Usability is to ensure that interactive products are easy to learn, effective to use and enjoyable from the user’s perspective. This can be achieved by optimizing the interaction between people and product to enable them to carry out their routine work. Among the usability goals identified are:

7.2.1 Efficiency

One of the goals in usability requirements is efficiency. One of the common tasks of the marketing planner is to consolidate the forecast submitted by the sales office. In the Forecast Sales Estimate (FSE) tab, user is given an option to choose the drop down list of the sales office. By doing this allows user to choose any sales office in order to view the forecast submitted. But if no sales office is selected, then all the forecast submitted by all the sales office will be displayed. This will help user to consolidate the forecast easily submitted by the sales offices. Thus, the system manages to support users in carrying out their common tasks.
7.2.2 Learnability

Learnability refers to how easy a system is to learn to use. Even though the user of the system can be categorized as expert user, but people do not like to spend hours in front of the computer to learn to use the system. In CRMS, the interface of the system is design in such a way that it simple so that it is easier for user to learn to use it. The basic mode which is Add, Save and Delete are all available in all the screens. This is done so that once the user knew to use the modes in the first screen, then he/she can apply it in other screens. This thus saves the learning time so that user can use some other quality time to learn something new in the system.
Figure 7.13: Basic Modes in CRMS

Besides this, the tool tip is also one of the features available in the system to help the user to learn faster about the system. Tool tip is a standard Windows control that provides a small pop-up window with descriptive text. The use of the tool tip text on the buttons in the toolbar is shown in the properties box of the toolbar as shown below:

Figure 7.14: Tool Tip Settings
As shown in Figure 7.13, the tool tip text for New button is set to ‘Add New Record’. So whenever user places a cursor on the New button for a while, a small window will pop up and shows ‘Add New Record’. Each button on the toolbar have its own tooltip (refer Table 7.2) By providing the feature, then the first timer user will be able to learn faster about the use of the standard toolbars.

<table>
<thead>
<tr>
<th>Button</th>
<th>Tool tip Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Add New Record</td>
</tr>
<tr>
<td>Save</td>
<td>Save Current Record</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete Current Record</td>
</tr>
<tr>
<td>Find</td>
<td>Find An existing Record</td>
</tr>
<tr>
<td>Info</td>
<td>Current Record Information</td>
</tr>
<tr>
<td>Cancel</td>
<td>Cancel Current Operation</td>
</tr>
<tr>
<td>Close All</td>
<td>Close Current Module</td>
</tr>
<tr>
<td>Print</td>
<td>Printing Report</td>
</tr>
<tr>
<td>Help</td>
<td>Help On Current Module</td>
</tr>
<tr>
<td>Exit</td>
<td>Exit CRMS</td>
</tr>
</tbody>
</table>

7.2.3 Memorability

Memorability refers to how easy the system is remembered to be use. Since the system will be use only once a month, therefore it is design to be interactive, simple and easy to remember so that user do not have to recalled on how to use the system each time they want to consolidate the forecast every month. The sequence of the moderation process had been incorporated in a single screen, in three separate tabs in order.
Figure 7.15: Moderation Processes Screen
7.2.4 Utility

Utility refers to the extent to which the system provides the right kind of functionality so that user can do what he/she need or want to. Among of the good utilities available in the system are:

a. Switching company

Figure 7.16: Change Company Submenu

The option to switch company is available in the Administration menu. By clicking in the Change Company submenu, user will be directed to the Change Company screen and they can easily switch to other granted companies without having a hassle to exit the system and re-login into the system. These thus save the user’s time and effort.

Figure 7.17: Company Selection Screen
The coding of the switching company process is shown below:

01Private Sub mnuChangeCompany_Click()
02   Dim msg$, tmp As Integer
03   If Forms.Count > 1 Then
04      msg$ = “All open forms will be closed.” & vbLf & vbLf
05      msg$ = msg & “Do you wish to proceed?”
06      Beep
07      tmp = MsgBox(msg$, vbYesNo + vbQuestion, “Change Company”)
08      If tmp = vbNo Then Exit Sub
09      MousePointer = vbHourglass
10      ShowMsg “Please wait for the form(s) to close...”
11      Do Until (ActiveForm Is Nothing)
12         Unload ActiveForm
13      Loop
14      MousePointer = vbDefault
15      End If
16      ExitType = False
17      Unload Me
18      frmCompany.Show vbModal, Me
19      Me.Show
20 End Sub

Line 04 shows that the system alerts the user that all the open form will be closed and they is given an options whether they want to continue or not. If they wants to continue, Line 10 shows that the system informs the user to wait for the form to be close. Line 18 shows that the Change Company screen is shown to allow user to switch to another company.
b. Change Password

Users in CRMS are given an access to change the password on their own. When they choose Change Password from Administration menu (refer Figure 7.18), then the Change Password screen is displayed (refer Figure 7.19).

![Figure 7.18: Change Password Submenu](image1)

![Figure 7.19: Change Password Screen](image2)

Since each user id in CRMS is using Oracle username, each time the password is changed, then the Oracle password also is changed automatically. This is supported by the following coding:

```vbnet
01 Private Sub cmdChange_Click()
02 Dim sql$ As Object
03 Dim Rec As Object
04 Dim intProfileNo As Integer

05 If Len(txtNewPwd) > 0 Then
06 If Len(txtConfirmPwd) > 0 Then
07 If Ucase$(Trim$(txtNewPwd)) = Ucase$(Trim$(txtConfirmPwd)) Then
```

182
5.3.4.2 sql = “alter user “ & UserObj.UserID & _
“ identified by “ & Trim$(txtNewPwd)

09            On Error Resume Next
10            UserObj.RunSQL sql
11            If Err.Number <> 0 Then
12               Beep
13               MsgBox “Password contains invalid characters!” , vbCritical
14               txtConfirmPwd.Text = “”
15               If txtNewPwd.Visible Then txtNewPwd.SetFocus
16               Exit Sub
17            End If
18            On Error GoTo 0
19            Beep

20            sql = “Password changed successfully for user “ & UserObj.UserID & vbLf & vbLf
21            sql = sql & “The new password will take effect at next login.”
22            MsgBox sql, vbInformation, “Password Changed”
23            Unload Me
24            Else
25               Beep
26               MsgBox “New Password and Confirmed Password does not match!” , vbExclamation
27               txtConfirmPwd.Text = “”
28               If txtNewPwd.Visible Then txtNewPwd.SetFocus
29            End If
30            Else
31               Beep
32               MsgBox “Please confirm your password!” , vbExclamation
33               If txtConfirmPwd.Visible Then txtConfirmPwd.SetFocus
34            End If
35            Else
36               Beep
37               MsgBox “Please enter your password!” , vbExclamation
38               If txtNewPwd.Visible Then txtNewPwd.SetFocus
39            End If

40End Sub

Line 08 shows the SQL to change Oracle password.
7.3 OTHER CHARACTERISTIC OF GOOD GUI

Important characteristic of good GUI plays an important role in determining its success. Most of successful GUI’s share common characteristics. They are:

7.3.1 Sorting

Sorting feature is available in the system as it gives flexibility to user to sort the column based on their preferences.

Figure 7.20: Sorting in Feature Screen

01 Private Sub lvwFeature_ColumnClick( ByVal ColumnHeader As MSComctlLib.ColumnHeader) 
02   If lvwFeature.SortKey = ColumnHeader.Index – 1 Then 
03      If lvwFeature.SortOrder = lvwAscending Then 
04         lvwFeature.SortOrder = lvwDescending 

Whenever user double click on the list view (in this case Feature list view), the ColumnClick event for the list view will be triggered. When user click once on the Feature Code, the list will be order descending (refer Line 03, 04), and when user click for the second time, the list will be sorted ascending (refer Line 06). When the form is initially loaded, the list is sorted ascending as requested by user.

All the master screens, such as Feature, Main Contributor, User Access which have the list view have the utility to sort the list.

7.3.2 Font

One of the guidelines details on using the font is their usage should only use up to three fonts. The use of the font in CRMS is limited to only one type, which is Verdana.
The font size use is 8 and is used consistently throughout the screens. The font and its size are set in the properties box of the controls in the screen, as shown below:

![Figure 7.21: Setting Up Font in Property Box](image)

Knowledge intermittent users need simple, organized and well labeled display which will guide their actions.

### 7.3.3 Visualization

The populating of tyre sizes based on the popularity is done using Microsoft chart, which is build-in inside the Visual Basic 6.0. This is shown in the populateGraph module:

```vbnet
1. Private Sub populateGraph()
2. Dim idate As Date
3. Dim column, row As Integer
4. Dim ShiftSet As Object
5. Dim Booked As New Collection
6. Dim Booked2 As New Collection
7. Dim exist As Boolean
8. Dim Book As String
9. Dim Overtime As Boolean
10. Dim sql As String
11. Dim sqlset As Object
12. Dim strsql As String
13. Dim tempset As Object
14. Dim strItem As String
```
15. Dim I As Integer
16. Dim intcount As Integer
17. Dim intQuota As Integer

18. chTB.Visible = True

19. If Len(Trim(txtForecast.Text)) = 0 Then
20. MsgBox “The Forecast field is empty. Please enter the Forecast value”, vbExclamation
21. txtForecast.SetFocus
22. Exit Sub
23. End If

24. If Len(Trim(txtQuota.Text)) = 0 Then
25. MsgBox “The Quota field is empty. Please enter the Quota value”, vbExclamation
26. txtQuota.SetFocus
27. Exit Sub
28. End If

29. MousePointer = vbHourglass

30. strItem = “”

31. Select Case cboGraph
32. Case “2D-Line”
33. chTB.chartType = VtChChartType2dLine
34. Case “2D-Bar”
35. chTB.chartType = VtChChartType2dBar
36. Case “3D-Line”
37. chTB.chartType = VtChChartType3dLine
38. Case “3D-Bar”
39. chTB.chartType = VtChChartType3dBar
40. End Select

41. chTB.Legend.Location.LocationType = VtChLocationTypeBottom

42. sql = “select count(distinct a.itemcode) count, sum(fse” & Trim(txtForecast.Text) & “)” & _
“ from crm_fse_m a, crm_item_m b” & _
“ where a.itemcode = b.itemcode” & _
“ and fse” & Trim(txtForecast.Text) & “ > 0” & _
“ and fsemonth = ‘’ & Format(dtpTransdate.Value, “mm”) & “’” & _
“ and fseyear = ‘’ & Format(dtpTransdate.Value, “yyyy”) & “’” & _
“ order by sum(fse” & Trim(txtForecast.Text) & “) desc”

43. Set sqlset = OraConn.OpenRecordset(sql)
44. If Not sqlset.EOF Then
45. intcount = sqlset.Fields(“count”).Value
46. Else
47. intcount = 0
48. End If
49. chTB.ColumnCount = Val(Trim(txtQuota.Text))
50. chTB.ColumnLabelCount = Val(Trim(txtQuota.Text))
51. chTB.RowCount = 1
52. chTB.RowLabelCount = 1
53. chTB.row = 1
54. chTB.column = 1
55. idate = dtpTransdate.Value
56. row = 1
57. intQuota = 0
58. sql = “select a.itemcode, itemname, sum(fse” & Trim(txtForecast.Text) & “)” & _
“ from crm_fse_m a, crm_item_m b” & _
“ where a.itemcode = b.itemcode” & _
“ and fse” & Trim(txtForecast.Text) & “ > 0” & _
“ and fsemonth = ‘’ & Format(dtpTransdate.Value, “mm”) & “’” & _
“ and fseyear = ‘’ & Format(dtpTransdate.Value, “yyyy”) & “’” & _
“ group by a.itemcode, itemname” & _
“ order by sum(fse” & Trim(txtForecast.Text) & “) desc”
59. Set sqlset = OraConn.OpenRecordset(sql)
60. For column = 1 To intcount – 1
61. If intQuota < Val(Trim(txtQuota.Text)) Then
62. strSQL = "select sum(fse) as fse from crm_fse_m a" & _
   " where companycode = '" & strCompcode & "' & _
   " and fsemonth = '" & Format(dtpTransdate.Value, "mm") & "' & _
   " and fseyear = '" & Format(dtpTransdate.Value, "yyyy") & "' & _
   " and itemcode = '" & sqlset.Fields("itemcode").Value & "'
63. Set tmpset = OraConn.OpenRecordset(strsql)
64. chTB.column = column
65. chTB.Data = tmpset.Fields("fse").Value
66. chTB.ColumnHeader = sqlset.Fields("itemcode").Value
67. chTB.RowLabel = Format(idate, "mmm-yyyy")
68. intQuota = intQuota + 1
69. sqlset.MoveNext
70. End If
71. Next column
72. chTB.Visible = True
73. chTB.ShowLegend = True
74. chTB.Refresh
75. Set Booked = Nothing
76. MousePointer = vbDefault
77. Exit Sub
78. End Sub

The declaration of all the variables used in the module is defined in codes of **Line 2** until **Line 17**.

**Line 31 to 40** shows the choosing of graph type by the user is assigned to the Microsoft chart.
**Line 19 to 23** shows the validation of forecast values. If no value is keyed in, then system will prompt and error message and the populating of graph are not done until a value is keyed in.

**Line 24 to 28** shows the validation of quota values. If no value is keyed in, then system will prompt and error message and the populating of graph are not done until a value is keyed in.

**Line 42** is a SQL to check the number of size exist in the forecast table for the selected month and year. The number of record is assigned to the variable, intcount in **Line 45 or 47**.

**Line 58** is a SQL to retrieve all the size available in the selected month and year based on the popularity of the size in which the highest values will be sorted ascending.

**Line 62** is a SQL to retrieve the total number of forecast values for the size and selected month and year.

The number of forecast values is assigned to the Microsoft charts data is in **Line 65**.

### 7.3.4 Speed or responsiveness

The speed can be a make or break factor in determining the application whether it can be accepted in the user’s community. For example is the system performance is very slow in the online application, the user might want to abandon the system. The appearance of speed is done in several ways in the CRMS. They are:
a. Validation of screen

In CRMS, the validation of a whole screen is done on a whole screen basis and not on field-by-field basis. The validation based on field-by-field basis can slow down the performance of the system and some of the users are irritated by the message box prompted by the system while they are entering data into the system. This is supported in the following codes in save record module in Contributor screen.

```vbnet
01 Public Sub saveRecord() 
02    Dim iResp As VbMsgBoxResult 
03    Static i% 

04    If Trim$(txtContributorCode.Text) = "" Then 
05        MsgBox "Please enter data ", vbExclamation 
06        Exit Sub 
07    End If 

08    If Trim$(txtContributorDesc.Text) = "" Then 
09        MsgBox "Please enter data ", vbExclamation 
10        Exit Sub 
11    End If 

.. 

In the save Record module, validation is done from Line 04 until 07 and Line 08 until 11. In Line 04 to 07, validation is done to the contributor code to check whether user had input a data inside the field. If no data found, the system will prompt a message box, where user need to fill up the contributor code before the system can save the data. In Line 08 to 11, validation is done on the contributor name to validate the existence of contributor name.

b. Give power to user to key in data rapidly
In CRMS, users are given feature to key in the data rapidly by using the tab key. All the fields in each screen are order by tab index. All users have to do is by hitting the tab key and the cursor will move from one field to another fields based on the sequence of the fields on the screen. The tab index of each field is set in the property box of each control, as below:

![Figure 7.22: Setting up of Tab Index in Property Box](image)

**Figure 7.22: Setting up of Tab Index in Property Box**
As shown in the property box, the tab index of weight field in contributor screen had been set to 2 and the tab index of contributor code and contributor name had been set to 0 and 1 respectively. These will faster the input of the data since user is using keyboard to key in the data and they do not have to use the mouse to move the cursor to another fields.
c. Use of short cut key

The use of short cut key is made available in CRMS whenever users want to navigate through the menu. The short cut key is set in the property of the menu editor as follows:

![Menu Editor Interface]

**Figure 7.23: Setting Up the Short Cut in Menu Editor**

As shown in the property box of menu editor, the short cut key of submenu Contributor is set to Control + C. When users want to open the Contributor screen, they have to press button Control and C at the same time to open the screen. The entire submenu had been assigned with short cut key for easier navigation in CRMS.
Table 7.3: Short cut key list in CRMS

<table>
<thead>
<tr>
<th>Menu</th>
<th>Short cut key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributor</td>
<td>Control + C</td>
</tr>
<tr>
<td>Feature</td>
<td>Control + F</td>
</tr>
<tr>
<td>Product Group</td>
<td>Control + G</td>
</tr>
<tr>
<td>User Access</td>
<td>Control + U</td>
</tr>
<tr>
<td>Exit</td>
<td>Control + X</td>
</tr>
<tr>
<td>User Case</td>
<td>Control + S</td>
</tr>
<tr>
<td>Moderation Process</td>
<td>Control + M</td>
</tr>
<tr>
<td>Factory Plan</td>
<td>Control + P</td>
</tr>
<tr>
<td>Change Password</td>
<td>Control + W</td>
</tr>
<tr>
<td>Change Company</td>
<td>Control + H</td>
</tr>
</tbody>
</table>

7.3.5 User’s Real World Metaphor

The real world metaphor should always be used whenever possible. The using of graphical representation can be immediately intuitive to users and help them to learn the application faster. In CRMS, the concept is used in icons on the toolbar, in which the icons used to represent the operations in the system, are similar to the icons used in Microsoft products.

![Icons in CRMS toolbar](image1)

![Icons in Microsoft Word 2003](image2)
7.3.6 Information Within Easy View

The information on each screen is organized and was designed to minimize for minimum movement of the head and eyes. As mentioned in Chapter 2 – Literature Review, during the daylight, human visual acuity is best at the front of the eye. This is supported in Marketing Moderation screen, where by all the columns in the grid are made within easy view and user does not need to use the scroll bar to scroll to the right or left. By looking straight, the whole information is available on the screen.

Figure 7.24: Marketing Moderation Screen
7.3.7 Colours

In CSTPJ, the corporate colours used are orange and yellow. Thus this had been incorporated inside the system whereby the yellow colour had been used as highlighting colour in Marketing Moderation screen (refer figure 7.24). Only two colours are used to alert users which are red and yellow. This is supported by the fact that, human eyes are sensitive to small changes in red colour. The use of too many colours is useless as only eight to ten different colours managed to be identified accurately.

![Customer Relationship Management System](image)

**Figure 7.25: Use of Red Colour to Prompt User**

This is supported in the following codes in login click event in Login screen:

```vbscript
01 Private Sub cmdLogin_Click()
02   Dim lpAppName As String, lpKeyName As String
03   Dim lpFileName As String
04   Dim X As Long
05   Dim strUserPass As String
06   If Len(Trim(txtUserID.Text)) = 0 Then
07      MsgBox "The User ID is empty. Please enter your User ID", vbExclamation
08      txtUserID.BackColor = vbRed
09      txtUserID.SetFocus
10   Else
11      X = &H80000005
12      txtUserID.BackColor = &H80000005
13   End If
```
If Len(Trim(txtPassword.Text)) = 0 Then
    MsgBox "The Password is empty. Please enter your Password", vbExclamation
    txtPassword.BackColor = vbRed
    txtPassword.SetFocus
    Exit Sub
Else
    txtPassword.BackColor = &H80000005
End If

At **Line 06**, system checks for the length of the UserID. If no userid is keyed-in, the system will prompt an error message to the user and the field of UserID will change colour to red (**Line 08**). At **Line 14**, system will check against the password. And same as the UserID, if no password is keyed-in, system will prompt an error message at **Line 15** and the field of password will change to red colour.

### 7.4 CASE BASE REASONING (CBR)

When the system receives the input problem from the user, the identify module extracts the feature of the problem which later is used in matching module. In order for the match rule to understand the problem, each feature of the problem is structured and is assigned with the indexing rule.

The indexing rule consists of assigning:

1. Indices ID for each ContributorCode, feature list and case.
2. Weight for ContributorID
3. Numeric Value for feature list.
For this system, it provides choices for the user to choose in order to help user identify and indicate his problem. It is easier for the system to understand the user’s problem when every feature is arranged in well structure index and is assigned with a weight and numeric value. The value that is assigned to each feature is used for indexing. This indexing method is known as the checklist-based indexing. It is a method in which the domain is analyzed and the dimensions that tend to be important are computed. These are put in a checklist and all cases are indexed by their values along these dimensions.

For the identify module, it consists of three modules. The modules are the structure indexing module, assign weight module and assign numeric value module.

7.4.1 Structure Indexing Module

In this section detail of the structure indexing module is explained. It consists of the important part of the coding, explanation of the coding and the interface that is related with this module.

```vba
Private Sub populateFeature(strObject As Integer)
    Dim sql As String
    Dim Rec As Object
    Dim strCont As String
    Dim sText As String

    Select Case strObject
    Case 1
        strCont = "10"
        cboStock.RemoveAll
    Case 2
        strCont = "20"
        cboPast.RemoveAll
    Case 3
```

199
strCont = "30"
cboDemand.RemoveAll
Case 4
strCont = "40"
cboFactory.RemoveAll
Case 5
strCont = "50"
cboHistory.RemoveAll
End Select

sql = "select featurecode, featurename, weight, numvalue" & 
     " from crm_feature_m a, crm_contributor_m b" & 
     " where a.contributorcode = " & strCont & 
     " and a.contributorcode = b.contributorcode" & 
     " order by featurecode"
Set Rec = OraConn.OpenRecordset(sql)
Do While Not Rec.EOF
    sText = "" & Rec.Fields("featurecode").Value & "," & "" & "" & 
    Select Case strObject
    Case 1
cboStock.AddItem sText
    Case 2
cboPast.AddItem sText
    Case 3
cboDemand.AddItem sText
    Case 4
cboFactory.AddItem sText
    Case 5
cboHistory.AddItem sText
    End Select
    Rec.MoveNext
Loop
End Sub

The input problem descriptor from user is change into structured way by giving choices for the user to choose to help user identify his problem. From Line 01 to Line 40, the user is given with a list of option which consist information about the:
a. Stock  
b. Past month sales  
c. Demand  
d. Factory capacity  
e. Popularity  

The user just needs to choose only one answer for each feature display. When the interface for user is created, the input problem descriptor is structured and labeled in the same way as the information in the old cases and can be easily understood by the system.

Figure 7.26: Input Problem Descriptor Screen
7.4.2 Assign Weight Module

In this section detail of the assign weight module is explained. It consists of the important part of the coding, explanation of the coding and the interface that is related with this module.

Sub AssignValue()
On Error GoTo errHandler

With Contributor.Contributor(Trim$(txtContributorCode))
.ContributorCode = "" & Trim$(txtContributorCode.Text)
.ContributorName = "" & txtContributorDesc.Text
.Weight = "" & Val(txtWEight.Text)
End With

Exit Sub

errHandler:
Debug.Print Err.Number & Err.Description
MsgBox Err.Number & Err.Description

End Sub

The above coding is extracted from the weight module which represents the weight assigning procedure. At Line 04 to 06, the values keyed in at Contributor Code, Contributor Name and the weight are stored. Below is the table (Table 7.4) describes the Contributor Code, the name for each Contributor Code and the weight assigned for each contributor.

Table 7.4: Weight Description Table

<table>
<thead>
<tr>
<th>ContributorCode</th>
<th>Contributor Name</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Stock</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>Past month sales</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>Demand</td>
<td>4</td>
</tr>
<tr>
<td>40</td>
<td>Factory capacity</td>
<td>4</td>
</tr>
<tr>
<td>50</td>
<td>Popularity</td>
<td>10</td>
</tr>
</tbody>
</table>
Weight indicates the different level of importance for each feature. These weights are very important in the calculation for the most similar case with the present case. In order for the system to provide the right solution, the weight of the popularity must be higher than other features and this is the main reason the purpose of implementing weight. From the researcher observation and research, the weight 10 is assigned to the virus type because this feature is the most important feature that contributes for the selection of the accurate or similar solution.

The weight of the virus type is fixed to 10 and the weight of other features is fixed to 4. The reason why the other features are fixed to 3 is to ensure the popularity contributes majority value in global similarity. Global similarity represents the total of the local similarity for all the input descriptor. It is the similarity on case or abject level where it combines local similarity measures and takes care of different important of feature weight. Local similarity represents the weight of the particular feature multiply with the numeric value for the particular feature. It is also known as the similarity on feature level.
7.4.3. Assign Numeric Value Module

In this section detail of the assign numeric module is explained. It consists of the important part of the coding, explanation of the coding and the interface that is related with this module. The value for each particular numeric value can be referred at Appendix E.

```vba
01 Sub AssignValue()
02 On Error GoTo errHandler

03 With Feature.Feature(Trim$(txtFeatureCode.Text))
04 .FeatureCode = Trim$(txtFeatureCode.Text)
05 .FeatureName = Trim$(txtFeatureName.Text)
06 .ContributorCode = cboContributorCode.Text
07 .Value = txtValue.Text
08 End With
```

Figure 7.27: Weight Assign Screen
The above coding is extracted from the numeric value module which represents the numeric value assigning procedure by the admin. User is only allowed to choose the answer selection which represent the numeric value which for new case in the similarity module. From Line 04 to 07, the appropriate values of featurecode, featurename, contributorcode and the numeric value had been assigned accordingly. The numeric value ID for the numeric value description is as illustrated in Table 7.5. The reason why the ID is given as illustrated in the table, to enable to identify the ID based on the Group ID easily. The details and specific numeric value ID with the numeric value and description can be referred at Appendix C.

Table 7.5: Numeric Value ID Table

<table>
<thead>
<tr>
<th>Numeric Value ID</th>
<th>Description</th>
<th>Contributor Name</th>
<th>Contributor Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>101-200</td>
<td>Stock specific feature</td>
<td>Stock</td>
<td>10</td>
</tr>
<tr>
<td>201-300</td>
<td>Past month sales specific feature</td>
<td>Past month sales</td>
<td>20</td>
</tr>
<tr>
<td>301-400</td>
<td>Demand specific feature</td>
<td>Demand</td>
<td>30</td>
</tr>
<tr>
<td>401-500</td>
<td>Factory capacity specific feature</td>
<td>Factory capacity</td>
<td>40</td>
</tr>
<tr>
<td>501-600</td>
<td>Popularity specific feature</td>
<td>Popularity</td>
<td>50</td>
</tr>
</tbody>
</table>

The numeric value is assigned at the range 1 to 20 for the specific or sub features for each group. The range is set from 1 to 20 to allow each input descriptor to show the different and similarity among themselves. If the range is set lower than 10, it is difficult
to show the different and similarity among the input descriptor because all the numeric value are to close.

![Figure 7.28 Numeric Value Interface](image)

### 7.4.4 Search Module

In this section detail of the search module is explained. It consists of the important part of the coding, explanation of the coding and the interface that is related with this module.

```vbnet
01dblCompare1 = grpDiv1
02    dblCompare2 = grpDiv2
03    sql = "select * from crm_caseheader_t a, crm_moderate_m b" & _
04        " where casemonth = " & Format(dtpTransDate.Value, "mm") & ",&" & _
05        " and caseyear = " & Format(dtpTransDate.Value, "yyyy") & ",&" & _
```
Once the input problem descriptor is received by the system, the system searches a case from the case base that match the index given in identify module. It is straightforward process. The system searches based on the index and if there is an exact matching of the retrieved case, the solution without any adaptation is displayed. Otherwise, if the system did not find the exact matching the indexes of the problem are passed to the similarity module.
7.4.5 Similarity Module

In this section, the detail of the similarity module is explained. It consists of the important part of the coding, explanation of the coding and the interface that is related with this module. This module is the most important feature for this system. If there is no exact match case found in the search module, the similarity module finds other similar cases from the case base based on the assigned index of the input case. The method used to find the similarity cases is the nearest neighbour algorithm. This method involves the assessment of similarity between stored cases and the new input case based on matching a weighted sum of features. The calculation of similarity only can be carried out using integer but the data input for each problem descriptor by the user in text type, therefore each selection of problem descriptor is changed to the integer type. The method to solve this problem is to assign a numeric value to each problem descriptor. Therefore the numeric value assigned to these problem descriptors is unique to each case and determine which case is retrieved from similarity algorithm. The numeric value that is assigned to each problem descriptor is based on the important of the feature to the domain knowledge is given similar which is based on the researcher research about viruses at Chapter 2. Similarity algorithm used for this system is nearest neighbour. Similar problem description is given with the similar numeric value. The similarity of these problem descriptors are based on their features which contribute to the problem solution. The formula of the nearest neighbour algorithm is already explained in Chapter 2. The calculation for the similarity of each feature involves multiplication of weight and similarity function. Main calculation for the each local similarity as shown below:

$$\sum_{i=1}^{n} W_i \times \text{sim} \left( f_i^L, f_i^R \right)$$
$W_i$ is the weight of the particular feature and $\text{sim}$ is the similarity function and $f_i^I$ and $f_i^R$ are the values for feature $i$ in the input and retrieved cases respectively.

Every feature in the input case is matched to its corresponding feature in the stored or old case, the degree of match of each pair is computed and based on the importance assigned to each dimension, an aggregate match score or also known as the local similarity is computed. For each feature in the input case:

- The corresponding feature in the stored case is identified
- Compare the two values to each other and compute the degree of match
- Multiply by a coefficient representing the importance of the feature to the match

Later the differences of new numeric value and the existed numeric value in the case base are counted. There are two ways how the numeric value is assigned. The first way is if the numeric value of that particular problem descriptor exactly the same with the numeric value of previous case, the value returned to the function is maximum. The maximum value is 20 and this is same with the range of the numeric value assigned to each problem descriptor which helps to simplify the calculation of the similarity function.

The second way is if the numeric value of that particular problem descriptor is different with the numeric value of previous case, the value returned to the function is maximum with the subtraction of these two numeric values. As coded in the above stated line, the general formula used is:
\[ \text{sim} \left( f_i^I, f_i^R \right) \]

\[ \text{different} = f_i^I - f_i^R \]

if different > 0 then result = range of value - different
else if different < 0 then result = different – range of value

local similarity = result * weight

Then all the results which consist of the multiplication of the weight and the differences of the numeric value are added to derive the local similarity. The local similarity represents the aggregate degree of match of the old case to the new input. The global similarity is then calculated. Global similarity is the total of all the local similarity for all features in a case. The similarity module task found a case from the case base that match the value of the global similarity. If there is an exact matching of the retrieved case with the global similarity value, it means this case is the best solution for the user as all the features in that case match the need of the user. Later this case is sent to the display module in order to display the solution to the user.

Otherwise, a case can be chosen by choosing the item with the largest score. The three biggest global similarities are chosen from the case base. The new global similarity is compared with each global similarity of the cases in the case base. Three variables used to record the three biggest global similarity and three variables used to record the refer key of that case. If the new global similarity greater than the biggest global similarity, the new global similarity becomes the new biggest global similarity and the previous one is shifted to the second biggest global similarity. Every time when the new global similarity
is instantiated, the refer key of case called “ID” is also recorded. Later the 3 best matched cases are sent to the display module in order to display the solution to the user.

7.5 MODULES IMPLEMENTATION

7.5.1 Sign up Module

Access authentication module authenticates the accesses and requests users. Table 7.6 describes the implementation of the sign up module as elicited in the requirement list in Chapter 5 based on the Requirement ID provided.

![Figure 7.29: Sign up screen](image)

Table 7.6: Verification of Sign up module

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a registration interface to allow system administrator to register the personal details.</td>
<td>CRM-Mod01-01</td>
<td>Done (refer figure 7.29)</td>
</tr>
</tbody>
</table>
2. Each registration screen of the system will consist of a user id, employee name, network id and password.  

3. The system will ensure the user id created is uniquely named after the network id.  

5. The system will verify the input data before storing the data in the database by triggering an alert to the user if the value is invalid.

<table>
<thead>
<tr>
<th></th>
<th>Each registration screen of the system will consist of a user id, employee name, network id and password.</th>
<th>CRM-Mod01-02</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The system will ensure the user id created is uniquely named after the network id.</td>
<td>CRM-Mod01-03</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>The system will verify the input data before storing the data in the database by triggering an alert to the user if the value is invalid.</td>
<td>CRM-Mod01-04</td>
<td>Done</td>
</tr>
</tbody>
</table>

Figure 7.30: Verification Message Box

7.5.2 Access Authentication module

Access authentication module authenticates the accesses and requests users. **Table 7.7** describes the implementation of the access authentication module as elicited in the requirement list in **Chapter 5** based on the Requirement ID provided.
Table 7.7: Verification of Access Authentication module

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a login interface that consists of the username, password and database fields.</td>
<td>CRM-Mod02-01</td>
<td>Done (refer figure 7.25)</td>
</tr>
<tr>
<td>2.</td>
<td>Both inputs shall be verified first by the system in order to authenticate the user before entering the system.</td>
<td>CRM-Mod02-02</td>
<td>Done (refer figure 7.31)</td>
</tr>
<tr>
<td>3.</td>
<td>The system will verify and authenticate the screen accesses made by the authorized user.</td>
<td>CRM-Mod02-03</td>
<td>Done (refer figure 7.31)</td>
</tr>
</tbody>
</table>

Figure 7.31 Connection Error Message Box

7.5.3 Consolidate Forecast Module

Consolidate Forecast module allows the marketing planner to summarize the forecast details which were input from the sales representative. Table 7.8 describes the implementation of the consolidate forecast module as elicited in the requirement list in Chapter 5 based on the Requirement ID provided.
### Table 7.8: Verification of Consolidate Forecast module

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a consolidate forecast interface that consists of the item code, item name, forecast month and the number of item forecast for the month.</td>
<td>CRM-Mod03-01</td>
<td>Done (refer figure 7.12)</td>
</tr>
<tr>
<td>2.</td>
<td>The system will listed all the sales office available in the company.</td>
<td>CRM-Mod03-02</td>
<td>Done (refer figure 7.12)</td>
</tr>
<tr>
<td>3.</td>
<td>The system will allow the selection of the sales office by the user.</td>
<td>CRM-Mod03-03</td>
<td>Done (refer figure 7.12)</td>
</tr>
<tr>
<td>4.</td>
<td>The system will display the consolidated forecast figures based on the next six month.</td>
<td>CRM-Mod03-04</td>
<td>Done (refer figure 7.12)</td>
</tr>
<tr>
<td>5.</td>
<td>The system will allow user to export the data to the Microsoft spreadsheet to allow them to manipulate data inside that tool.</td>
<td>CRM-Mod03-05</td>
<td>Done (refer figure 7.12)</td>
</tr>
<tr>
<td>6.</td>
<td>The system will allow the details to be printed.</td>
<td>CRM-Mod03-06</td>
<td>Done (refer figure 7.12)</td>
</tr>
<tr>
<td>7.</td>
<td>The system should prompt users for confirmation each time user wanted to perform the task.</td>
<td>CRM-Mod03-07</td>
<td>Done (refer figure 7.9)</td>
</tr>
<tr>
<td>8.</td>
<td>The system, as much as possible should provide informative content to user.</td>
<td>CRM-Mod03-08</td>
<td>Done (refer figure 7.10)</td>
</tr>
</tbody>
</table>
7.5.4 Populate Graph Module

Populate graph enables user to visualize the item based on the highest number of items forecasted. Table 7.9 describes the implementation of the populate graph module as elicited in the requirement list in Chapter 5 based on the Requirement ID provided.

Table 7.9: Verification of Populate Graph module

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a populate graph interface that consists of the item code and the number of item forecast for the month.</td>
<td>CRM-Mod04-01</td>
<td>Done (refer figure 7.32)</td>
</tr>
<tr>
<td>2.</td>
<td>The system will allow user to set the selection by allowing them to choose type of graph that will be display, the forecast month and the top number of item that will be display.</td>
<td>CRM-Mod04-02</td>
<td>Done (refer figure 7.32)</td>
</tr>
<tr>
<td>3.</td>
<td>The system shall gives flexibility to define the most popular items in the range of 1 to 100.</td>
<td>CRM-Mod04-03</td>
<td>Done (refer figure 7.32)</td>
</tr>
<tr>
<td>4.</td>
<td>The item shall be coded with different colours to make it easier to visualize.</td>
<td>CRM-Mod04-04</td>
<td>Done (refer figure 7.32)</td>
</tr>
<tr>
<td>5.</td>
<td>The system shall sort the item based on the highest number of item.</td>
<td>CRM-Mod04-05</td>
<td>Done (refer figure 7.32)</td>
</tr>
<tr>
<td>6.</td>
<td>The system will allow the details to be printed.</td>
<td>CRM-Mod04-06</td>
<td>Done (refer figure 7.32)</td>
</tr>
</tbody>
</table>
7. The system should prompt users for confirmation each time user wanted to perform the task.

<table>
<thead>
<tr>
<th>CRM-Mod04-08</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>(refer figure 7.9)</td>
<td></td>
</tr>
</tbody>
</table>

8. The system shall allow the graph to be enlarged.

<table>
<thead>
<tr>
<th>CRM-Mod04-09</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>(refer figure 7.32)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7.32: Visualization Screen
### 7.5.5 Moderate Forecast Module

Moderate Forecast module is to allow the marketing planner to moderate the forecast based on factors that had been defined in the Main Contributor module. Table 7.10 describes the implementation of the moderate forecast module as elicited in the requirement list in Chapter 5 based on the Requirement ID provided.

#### Table 7.10: Verification of Moderate Forecast module

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a moderate forecast interface that consists of the item code, item name, factory code, available stock, forecast number and the number of item forecast based on market.</td>
<td>CRM-Mod05-01</td>
<td>Done (refer figure 7.33)</td>
</tr>
<tr>
<td>2.</td>
<td>The system will highlight the number of forecast that is more than 1000, with yellow colour. This is for easier identification as these items are popular items and demanded more by the customers.</td>
<td>CRM-Mod05-02</td>
<td>Done (refer figure 7.33)</td>
</tr>
<tr>
<td>3.</td>
<td>The system shall sort the details by item code.</td>
<td>CRM-Mod05-03</td>
<td>Done (refer figure 7.33)</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Reference</td>
<td>Status</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
<td>---------</td>
</tr>
<tr>
<td>4</td>
<td>The system shall allow the marketing planner to modify the number of forecast by each market.</td>
<td>CRM-Mod05-04</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>(refer figure 7.33)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>The system should auto-calculate the total number of item upon changes made by the marketing planner.</td>
<td>CRM-Mod05-05</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>(refer figure 7.33)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>The system will allow user to export the data to the Microsoft spreadsheet to allow them to manipulate data using the tool.</td>
<td>CRM-Mod05-06</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>(refer figure 7.34)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>The system will allow the details to be printed.</td>
<td>CRM-Mod05-07</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>(refer figure 7.33)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>The system should allow user to use the same mode (new, save and delete) which is available in the other interfaces.</td>
<td>CRM-Mod05-08</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>(refer figure 7.33)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>The system should prompt users for confirmation each time user wanted to perform the task.</td>
<td>CRM-Mod05-09</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>(refer figure 7.9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>The system, as much as possible should provide informative content to user.</td>
<td>CRM-Mod05-10</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>(refer figure 7.10)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.5.6 Schedule Factory Plan Module

Schedule Factory Plan is a module to allow the factory planner to schedule the factory plan/program based on the moderated marketing figures. Table 7.11 describes the
implementation of the schedule factory plan module as elicited in the requirement list in

Chapter 5 based on the Requirement ID provided

d| No. | Requirement Description                                                                 | Requirement ID | Implementation |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a schedule factory plan interface that consists of rim size, the</td>
<td>CRM-Mod06-01</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>item code, item name, factory code, the number of moderated item based on market and</td>
<td></td>
<td>(refer figure 7.34)</td>
</tr>
<tr>
<td></td>
<td>scheduled items based on weeks.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>The system will display the number of moderated items based on market.</td>
<td>CRM-Mod06-02</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(refer figure 7.34)</td>
</tr>
<tr>
<td>3.</td>
<td>The system shall allow the factory planner to edit the moderated item on weekly basis.</td>
<td>CRM-Mod06-03</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(refer figure 7.34)</td>
</tr>
<tr>
<td>4.</td>
<td>The system should allow user to use the same mode (new, save and delete) which is</td>
<td>CRM-Mod06-04</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>available in the other interfaces.</td>
<td></td>
<td>(refer figure 7.34)</td>
</tr>
<tr>
<td>5.</td>
<td>The system should prompt users for confirmation each time user wanted to perform the task.</td>
<td>CRM-Mod06-05</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(refer figure 7.9)</td>
</tr>
</tbody>
</table>
6. The system, as much as possible should provide informative content to user.  

CRM-Mod06-06 Done  
(refer figure 7.10)

7. The system will verify the input data before to storing the data in the database by triggering an alert to the user if the value is invalid.  

CRM-Mod06-07 Done  
(refer figure 7.36)

---

**Figure 7.35: Factory Plan Screen**
7.5.7 Maintain Main Contributor Module

Main Contributor modules allows user to input the factors that contribute to decision making in moderation process. Table 7.12 describes the implementation of the maintain main contributor module as elicited in the requirement list in Chapter 5 based on the Requirement ID provided.

Table 7.12: Verification of Maintain Main Contributor Module

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a main contributor interface that consists of contributor code,</td>
<td>CRM-Mod07-01</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td>contributor name and weight.</td>
<td></td>
<td>(refer figure 7.37)</td>
</tr>
<tr>
<td>2.</td>
<td>The system shall allow user to sort the contributor listing based on their preference.</td>
<td>CRM-Mod07-02</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(refer figure 7.37)</td>
</tr>
<tr>
<td>3.</td>
<td>The system should only allow numeric values in Weight field.</td>
<td>CRM-Mod07-03</td>
<td>Done</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(refer figure 7.36)</td>
</tr>
<tr>
<td></td>
<td>The system should allow user to use the same mode (new, save and delete) which is available in the other interfaces.</td>
<td>CRM-Mod07-04</td>
<td>Done (refer figure 7.37)</td>
</tr>
<tr>
<td>---</td>
<td>------------------------------------------------------------------------------------------------------------</td>
<td>--</td>
<td>------------------</td>
</tr>
<tr>
<td>5.</td>
<td>The system should prompt users for confirmation each time user wanted to perform the task.</td>
<td>CRM-Mod07-05</td>
<td>Done (refer figure 7.9)</td>
</tr>
<tr>
<td>6.</td>
<td>The system, as much as possible should provide informative content to user.</td>
<td>CRM-Mod07-06</td>
<td>Done (refer figure 7.10)</td>
</tr>
<tr>
<td>7.</td>
<td>The system should allow the input of data to be done rapidly with the using of keyboard by following the sequence of field on the screen.</td>
<td>CRM-Mod07-07</td>
<td>Done (refer figure 7.22)</td>
</tr>
<tr>
<td>8.</td>
<td>The values for Weight should ranging from 1 to 10.</td>
<td>CRM-Mod07-08</td>
<td>Done (refer figure 7.36)</td>
</tr>
<tr>
<td>9.</td>
<td>The system will verify the input data before to storing the data in the database by triggering an alert to the user if the value is invalid.</td>
<td>CRM-Mod07-09</td>
<td>Done (refer figure 7.36)</td>
</tr>
</tbody>
</table>
7.5.8 Maintain Feature Module

Maintain feature module allows user to maintain the features attached to the contributor. Table 7.13 describes the implementation of the maintain feature module as elicited in the requirement list in Chapter 5 based on the Requirement ID provided.

Table 7.13: Verification of Maintain Feature Module

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a feature interface that consists of contributor code, feature code, feature name and numeric value.</td>
<td>CRM-Mod08-01</td>
<td>Done (refer figure 7.20)</td>
</tr>
</tbody>
</table>
2. The system shall allow user to sort the feature listing based on their preference.  
   CRM-Mod08-02 
   Done 
   (refer figure 7.20)

3. The system should only allow numeric values in Numeric Values field.  
   CRM-Mod08-03 
   Done 
   (refer figure 7.36)

4. The system should allow user to use the same mode (new, save and delete) which is available in the other interfaces.  
   CRM-Mod08-04 
   Done 
   (refer figure 7.20)

5. The system should prompt users for confirmation each time user wanted to perform the task.  
   CRM-Mod08-05 
   Done 
   (refer figure 7.20)

6. The system, as much as possible should provide informative content to user.  
   CRM-Mod08-06 
   Done 
   (refer figure 7.10)

7. The system should allow the input of data to be done rapidly with the using of keyboard by following the sequence of field on the screen  
   CRM-Mod08-07 
   Done 
   (refer figure 7.22)

8. The values for Numeric Values should ranging from 1 to 20  
   CRM-Mod08-08 
   Done 
   (refer figure 7.36)

9. The system should tied the contributor to feature list  
   CRM-Mod07-09 
   Done 
   (refer figure 7.20)

10. The system will verify the input data before to storing the data in the database by triggering an alert to the user if the  
    CRM-Mod07-10 
    Done 
    (refer figure 7.36)
The Change Password modules allows user to change the Oracle password for security purpose. The passwords are strings of characters used to authenticate the users. If the user manages to login successfully, the system will trust that they are valid user and grant them access to the data. Table 7.14 describes the implementation of the change password module as elicited in the requirement list in Chapter 5 based on the Requirement ID provided.

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The system will provide a Change Password interface that consists of old password and new password.</td>
<td>CRM-Mod09-01</td>
<td>Done (refer figure 7.9)</td>
</tr>
<tr>
<td>2.</td>
<td>The system will make sure the old password is the current password user is using.</td>
<td>CRM-Mod09-02</td>
<td>Done (refer figure 7.9)</td>
</tr>
<tr>
<td>3.</td>
<td>The system will make sure the old password and current password values matched.</td>
<td>CRM-Mod09-03</td>
<td>Done (refer figure 7.9)</td>
</tr>
</tbody>
</table>
4. The new passwords must be a minimum of six characters with at least one numeric character and one alpha character.

<table>
<thead>
<tr>
<th>CRM-Mod09-04</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>(refer figure 7.9)</td>
<td></td>
</tr>
</tbody>
</table>

7.6 CONCLUSION

The coding for each module for this system and the interfaces that is related with the coding is discussed in this chapter. Each module is explained in structured way. This is to show the implementation of CRM concept, HCI and CBR approach inside the system.
CHAPTER 8
SYSTEM TESTING AND EVALUATION

8.0 INTRODUCTION

System testing and evaluation of the system is done in order to evaluate the system’s compliance with its requirements. There are few types of software testing, which can be divided as below.

8.1 UNIT TESTING (WHITE BOX TESTING)

It is a stage whereby each individual component is tested. The main objective in this stage is to focus on the small segment of code and aim to exercise a high percentage of internal paths. But the disadvantage of the testing is the tester may be biased by previous experience and the test value may not cover all the possible scenarios in the real life. The purposes of the testing are as the following:

a. to verify that the testing application forms are performing correctly
b. to ensure that all the required fields and buttons exist and working
c. to ensure that the flow of information and data entry is logical and correct based on the application’s business development
d. to verify the navigation between forms
8.1.1 Login Screen

Objective of testing 8.1.1: To verify whether the valid input had been keyed in correctly in order to make a connection to the database.

![Image of Login Screen]

Figure 8.1: Login Screen

Table 8.1: Login Test Plan

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Key in User name at User ID field</td>
<td>System will prompt a message box if the field is empty as Figure 8.2.</td>
</tr>
<tr>
<td>2</td>
<td>Key in password at Password field</td>
<td>System will prompt a message box if the field is empty as Figure 8.3.</td>
</tr>
<tr>
<td>3</td>
<td>Key in the Database at database field</td>
<td>System will prompt error message if the field is empty as Figure 8.4.</td>
</tr>
<tr>
<td>4</td>
<td>Click on Login button</td>
<td>System makes connection to the database and proceeds to the Company Selection screen. If the connection failed, system will prompt a message box as Figure 8.5.</td>
</tr>
</tbody>
</table>

![Image of User ID Empty Message Box]

Figure 8.2: User ID Empty Message Box
Figure 8.3: Password Empty Message Box

Figure 8.4: Database Empty Message Box

Figure 8.5: Connection Error Message Box
8.1.2 Company Selection Company Screen

![Company Selection Screen](image)

**Figure 8.6: Company Selection Screen**

*Objective of testing 8.1.2:* The testing is done to verify the company access assigned to the user. The number of company listed is based on the number of companies granted to the user.

**Table 8.2: Company Selection Test Plan**

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Choose one company</td>
<td>The company selected will be highlighted as Figure 8.6.</td>
</tr>
<tr>
<td>2</td>
<td>Click OK button</td>
<td>The system proceed to the Main screen.</td>
</tr>
</tbody>
</table>
8.1.3 Main Menu

Objective of testing 8.1.3: The testing is done to verify the tool tip shown for each buttons in the screen.

Table 8.3: Main Menu Test Plan

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Place cursor on one of the button on the top of the screen</td>
<td>The tool tip for each button is shown as Figure 8.7</td>
</tr>
</tbody>
</table>

![CRM - CST ALOR SETAR (aix)](image)

Figure 8.7: Tool Bar
8.1.4 Contributor Screen

Objective of testing 8.1.4: The testing is carried out to verify the data input for the contributors. This is to ensure that each data are inserted, updated correctly in the case base.

Table 8.4: Contributor Test Plan

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click on Setup menu and click on Master and choose Contributor sub menu</td>
<td>The Contributor screen is shown as Figure 8.8.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Add mode:</strong> Click on New button</td>
<td>All the values on the screen are cleared.</td>
</tr>
<tr>
<td>3</td>
<td>Key in the Contributor Code</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Key in the Contributor Name</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Key in the Weight</td>
<td>The system prompts error message is the un-numeric value is keyed-in as Figure 8.9.</td>
</tr>
<tr>
<td>6</td>
<td>Click on Save button</td>
<td>The records will be saved and populated in the list view.</td>
</tr>
<tr>
<td>7</td>
<td><strong>Update mode:</strong> Choose any record from the list view</td>
<td>The details of the selected record are shown on the screen.</td>
</tr>
<tr>
<td>8</td>
<td>Update any values on the screen</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Click on Save button</td>
<td>The record is saved and the changes are shown on the list view.</td>
</tr>
<tr>
<td>10</td>
<td><strong>Delete mode:</strong> Choose any record from the list view</td>
<td>The details of the selected record are shown on the screen.</td>
</tr>
<tr>
<td>11</td>
<td>Click on Delete button</td>
<td>The confirmation message is shown to user. If user clicks Yes, then the record is deleted. If the user clicks on No, then the record will not be deleted</td>
</tr>
<tr>
<td>12</td>
<td><strong>View existing record</strong> Choose any record in the list view</td>
<td>The details of the selected record is shown on the screen</td>
</tr>
</tbody>
</table>
Figure 8.8: Contributor Screen

Figure 8.9: Numeric Value Message Box
8.1.5 Feature Screen

*Objective of testing* 8.1.5: The testing is carried out to test the data input for the feature.

This is to ensure that each data are inserted, updated correctly in the case base.

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click on Setup menu and click on Master and choose Feature sub menu</td>
<td>The Feature screen is shown as Figure 8.10.</td>
</tr>
<tr>
<td>2</td>
<td>Click on Contributor drop down list</td>
<td>The Contributor master list is shown on the drop down list.</td>
</tr>
<tr>
<td>3</td>
<td>Choose one contributor from the drop down list</td>
<td>The feature list is shown in the list view.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Add mode:</strong> Click on New button</td>
<td>All the values on the screen are cleared.</td>
</tr>
<tr>
<td>5</td>
<td>Key in the Feature Code</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Key in the Feature Name</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Key in the Numeric Value</td>
<td>The system prompts error message is the un-numeric value is keyed-in as Figure 8.9.</td>
</tr>
<tr>
<td>8</td>
<td>Click on Save button</td>
<td>The records will be saved and populated in the list view.</td>
</tr>
<tr>
<td>9</td>
<td><strong>Update mode:</strong> Update any values on the screen</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Click on Save button</td>
<td>The record is saved and the changes is shown on the list view.</td>
</tr>
<tr>
<td>11</td>
<td><strong>Delete mode:</strong> Choose any record from the list view</td>
<td>The details of the selected record is shown on the screen.</td>
</tr>
<tr>
<td>12</td>
<td>Click on Delete button</td>
<td>The confirmation message is shown to user. If user clicks Yes, then the record is deleted. If the user clicks on No, then the record will not be deleted.</td>
</tr>
<tr>
<td>13</td>
<td><strong>View existing records</strong> Choose any record in the list view</td>
<td>The details of the selected record is shown on the screen.</td>
</tr>
</tbody>
</table>
Figure 8.10: Feature Screen

8.1.6 User Case Screen

Screen: User case screen

Objective: This testing is carried out to test the user selection for the contributor. In order for the system to assign the numeric value to each of the decision descriptor, the user must give input for each of the decision descriptor. The testing is to ensure that the user choose an input for the decision descriptor.
Table 8.6: User Case Test Plan

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click on Transactions menu and click on Marketing and choose Use Case sub menu</td>
<td>The User Case screen is shown.</td>
</tr>
<tr>
<td>2</td>
<td>Click on Month drop down list to choose the Month of the transaction date</td>
<td>The calendar is shown.</td>
</tr>
<tr>
<td>3</td>
<td>Click on View button</td>
<td>The details are shown on the screen (if any).</td>
</tr>
<tr>
<td>4</td>
<td>Click on Stock drop down list</td>
<td>The Stock master list are shown in the drop down list.</td>
</tr>
<tr>
<td>5</td>
<td>Click on Past Month Sales drop down list</td>
<td>The Past Month Sales master lists are shown in the drop down list.</td>
</tr>
<tr>
<td>6</td>
<td>Click on Demand drop down list</td>
<td>The Demand master lists are shown in the drop down list.</td>
</tr>
<tr>
<td>7</td>
<td>Click on Factory Capacity drop down list</td>
<td>The Factory Capacity master lists are shown in the drop down list.</td>
</tr>
<tr>
<td>8</td>
<td>Click on History Trend drop down list</td>
<td>The History Trend master list are shown in the drop down list.</td>
</tr>
<tr>
<td>9</td>
<td>Click the cursor on the Moderation grid</td>
<td>The details of the Moderation are shown (if any)</td>
</tr>
<tr>
<td>10</td>
<td>Choose The Moderation code</td>
<td>The Moderation name is shown at Moderation name field</td>
</tr>
<tr>
<td>11</td>
<td>Click on Check button</td>
<td>System will list the decision of moderation (if any)</td>
</tr>
<tr>
<td>12</td>
<td><strong>Leaving empty values</strong>&lt;br&gt;Repeat step 1 – 3</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Leave the Stock as empty</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Leave the Past month sales as empty</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Leave the Demand as empty</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Leave the Factory Capacity as empty</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Leave the History Trend as empty</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Click on Check button</td>
<td>The system prompted errors as in the Figure 8.9, 8.11, 8.12, 8.13, 8.14 and 8.15.</td>
</tr>
</tbody>
</table>
Figure 8.11: Empty Stock Error Message

Figure 8.12: Empty Past Month Sales Error Message

Figure 8.13: Empty Demand Error Message

Figure 8.14: Empty Factory Capacity Error Message
8.1.7 Moderation Screen

Screen: Forecast Sales Estimate (FSE) tab - Moderation screen

Objective: The testing is carried out to check the functionality of the forecasting process in the screen.

Table 8.7: Forecast Test Plan

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click on Transactions menu and click on Marketing and choose</td>
<td>The Moderation screen is shown.</td>
</tr>
<tr>
<td></td>
<td>Moderation sub menu</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Click on Product group drop down list</td>
<td>The Product group master list is shown.</td>
</tr>
<tr>
<td>3</td>
<td>Click on Month drop down list to choose the Month of the transaction</td>
<td>The calendar is shown.</td>
</tr>
<tr>
<td></td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Click on Forecast Sales Estimate (FSE) tab</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Click on Sales office drop down list</td>
<td>The Sales office group master list is shown.</td>
</tr>
<tr>
<td>6</td>
<td>Click View button</td>
<td>The details of the forecast submitted by the sales office are shown on</td>
</tr>
</tbody>
</table>
Figure 8.16: Forecast Tab
**Screen**: Visualization tab - Moderation screen

**Objective**: The testing is carried out to check the visualization process in the screen.

**Table 8.8: Visualization Test Plan**

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click on Transactions menu and click on Marketing and choose</td>
<td>The Moderation screen is shown</td>
</tr>
<tr>
<td></td>
<td>Moderation sub menu</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Click on Product group drop down list</td>
<td>The Product group master list is shown</td>
</tr>
<tr>
<td>3</td>
<td>Click on Month drop down list to choose the Month of the transaction</td>
<td>The calendar is shown</td>
</tr>
<tr>
<td></td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Click on Visualization tab</td>
<td>The screen is as per Figure 8.17</td>
</tr>
<tr>
<td>5</td>
<td>Click on Display Graph drop down list</td>
<td>4 types of graph type are shown on the screen</td>
</tr>
<tr>
<td>6</td>
<td>Choose one type of graph</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Key in the forecast month that you want to view the graph</td>
<td>System will prompt message as Figure 8.9 if un numeric value is keyed in</td>
</tr>
<tr>
<td>8</td>
<td>Key in the Quota value</td>
<td>System will prompt message as Figure 8.9 if un numeric value is keyed in</td>
</tr>
<tr>
<td>9</td>
<td>Click on View button</td>
<td>The graph is populated</td>
</tr>
</tbody>
</table>
Figure 8.17: Visualization Tab
**Screen:** Marketing moderation tab - Moderation screen

**Objective:** The testing is carried out to check the functionality of the moderation process in the screen.

**Table 8.9: Moderation Test Plan**

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click on Transactions menu and click on Marketing and choose</td>
<td>The Moderation screen is shown.</td>
</tr>
<tr>
<td></td>
<td>Moderation sub menu</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Click on Product group drop down list</td>
<td>The Product group master list is shown.</td>
</tr>
<tr>
<td>3</td>
<td>Click on Month drop down list to choose the Month of the transaction</td>
<td>The calendar is shown.</td>
</tr>
<tr>
<td></td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Click on Marketing Moderation tab</td>
<td>The screen is as per Figure 8.18.</td>
</tr>
<tr>
<td>5</td>
<td>Click on View button</td>
<td>The details of the Marketing Moderation are shown. The popular items</td>
</tr>
<tr>
<td></td>
<td></td>
<td>are highlighted as per Figure 8.19.</td>
</tr>
<tr>
<td>6</td>
<td>Update the forecast values on the grid</td>
<td>Click on Save button and system prompts message box for confirmation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to save.</td>
</tr>
</tbody>
</table>
Figure 8.18: Marketing Moderation Tab
**Figure 8.19: Highlighted Rows**
8.1.8 Factory Plan Screen

*Screen:* Factory plan screen

*Objective:* The testing is carried out to check the functionality of the operations in the screen.

Table 8.10: Factory Plan Test Plan

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click on Transactions menu and click on Factory and choose Factory Plan sub menu</td>
<td>The Factory Plan screen is shown as Figure 8.20.</td>
</tr>
<tr>
<td>2</td>
<td>Click on Month drop down list to choose the Month of the transaction date</td>
<td>The calendar is shown</td>
</tr>
<tr>
<td>3</td>
<td>Click on View button</td>
<td>The details of the moderated FSE are shown (if any)</td>
</tr>
<tr>
<td>4</td>
<td>Key in the quantity to be produce on weekly basis (week by week)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Click on Save button</td>
<td>The record is saved</td>
</tr>
</tbody>
</table>
Figure 8.20: Factory Plan Screen
8.2 INTEGRATION TESTING

The integration testing can be divided into two sub-testing, which are called:

a. Top down integration testing

b. Bottom up integration testing

In this testing, the emphasis is stressed on module functionality and performance. By using this approach, errors in critical modules can be found earlier compared to top down integration testing where the errors in critical are found in late stage. This is the reason the bottom up integration testing approach is chosen while performing the integration testing for CRMS. The integration was performed in the system in order to test the flow of the master data setup in the Master screen, flows correctly to the Transaction screen.

<table>
<thead>
<tr>
<th>Master screens</th>
<th>Transaction screens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature screen</td>
<td>Use Case screen</td>
</tr>
<tr>
<td>Main Contributor screen</td>
<td>Use Case screen</td>
</tr>
<tr>
<td>Product Group screen</td>
<td>Moderation screen</td>
</tr>
<tr>
<td>User access screen</td>
<td>Login screen</td>
</tr>
<tr>
<td></td>
<td>Company selection screen</td>
</tr>
</tbody>
</table>

**Screen:** Feature and User Case screen

**Objective:** The testing is carried out to check the flow of data between Feature screen and User Case screen.

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click on Setup and choose Feature sub menu</td>
<td>The Feature screen is shown</td>
</tr>
<tr>
<td>2</td>
<td>Key in all the Feature details</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.12: Linkage from Feature screen to Use Case screen Test Plan
3 Click on Save button  
System prompted that the record has been saved

4 Click on Transactions menu and click on Marketing and choose User Case sub menu  
The User Case screen is shown

5 Click on Feature drop down list  
The new added feature is populated in the drop down list

Screen: Contributor and User Case screen

Objective: The testing is carried out to check the flow of data between Contributor screen and User Case screen.

Table 8.13: Linkage from Contributor screen to Use Case screen Test Plan

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click on Setup and choose Main Contributor sub menu</td>
<td>The Contributor screen is shown</td>
</tr>
<tr>
<td>2</td>
<td>Key in all the Main Contributor details</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Click on Save button</td>
<td>System prompted that the record has been saved</td>
</tr>
<tr>
<td>4</td>
<td>Click on Transactions menu and click on Marketing and choose User Case sub menu</td>
<td>The User Case screen is shown</td>
</tr>
<tr>
<td>5</td>
<td>Click on Contributor drop down list</td>
<td>The new added contributor is populated in the drop down list</td>
</tr>
</tbody>
</table>

Screen: User Access and Company Selection screen

Objective: The testing is carried out to check the flow of data between User Access screen and Company Selection screen.
Table 8.14: Linkage from User Access screen to Company Selection screen Test

Plan

<table>
<thead>
<tr>
<th>No</th>
<th>Steps</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Click on Setup and choose User Access sub menu</td>
<td>The User Access screen.</td>
</tr>
<tr>
<td>2</td>
<td>Assign new company to the user</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Click on Save button</td>
<td>System prompted that the record has been saved.</td>
</tr>
<tr>
<td>4</td>
<td>Login to the system</td>
<td>System validates the connection and proceed to Company Selection screen.</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>The new company assigned to the user is listed on the Company Selection list.</td>
</tr>
</tbody>
</table>
8.3 USABILITY TESTING

According to ISO 9241-11 (1998), usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. Usability testing focuses on determining if the product is easy to learn, satisfying to use and contains the functionality that the users desire. It is carried out at the various stages of design process. The usability testing had been carried out based on the following steps:

1. Developing a testing strategy
2. Planning a usability test
3. Setting up and configuring the usability lab
4. Identifying and recruiting the right users
5. Writing a test script
6. Running a usability test
7. Collecting and analyzing the results
8. Applying the results

The usability goals had been identified in the earlier stage, thus the usability testing had been carried out based on the goals set. The usability goals pre-set are:

i. Efficiency
ii. Learnability
iii. Memorability
iv. Utility
v. Other characteristic of good GUI which includes sorting, font, visualization, speed, user’s real world metaphor, information within easy view and colours.

The testing had been performed at different stages of the development cycle, such as prototyping stage, design stage and so on. All the feedback gathered from the session had been compiled and incorporated inside the system. Below is the final result compiled based on the questions asked in Appendix I – Usability Testing Questionnaires.

Figure 8.21: Result of Usability Testing
8.4 ACCEPTANCE TESTING

The purpose of the acceptance testing is to receive the feedback from the users regarding the system usability and the functionality of the system. It should be able to verify whether the final deliverables meet the user’s expectations. The CRMS user manual (refer to Appendix B – User Manual) had been given to the users and training had been conducted to explain in details on the system process and system functionality. Besides that, users are given chances to test the system based on the test plan given by the researcher. A set of questionnaires is also distributed to all the users (10 of them) in order to know their feedback on the system (refer to Appendix A – Questionnaires). Out of 11 questions, 5 of them are related to HCI, 3 questions related to CBR and another 2 question related to CRM. The rating of each questions are divided into 5 Lickert Scale:

1 – Totally Agree
2 – Partially Agree
3 – Neither Agree or Disagree
4 – Partially Disagree
5 – Totally Disagree
Below are the results of the feedback:

<table>
<thead>
<tr>
<th>Question 1</th>
<th>Does the sorting feature is useful in the system?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result 1</td>
<td>50% – Totally Agree</td>
</tr>
<tr>
<td></td>
<td>40% - Partially Agree</td>
</tr>
<tr>
<td></td>
<td>10% - Neither Agree or Disagree</td>
</tr>
<tr>
<td></td>
<td>0% - Partially Disagree</td>
</tr>
<tr>
<td></td>
<td>0% - Totally Disagree</td>
</tr>
</tbody>
</table>

**Chart 1**

- Totally Agree, 50%
- Partially Agree, 40%
- Neither Agree nor Disagree, 10%
- Partially Disagree, 0%
- Totally Disagree, 0%

**Analysis 1**

50% of the users are strongly satisfied that the sorting feature is helpful, while 40% partially agree with the questions and another 10% neither agree nor disagree. Therefore we can conclude that the sorting feature is one of characteristic of good GUI as it gives the flexibility to users to sort columns based on their preferences.

the utility easy to use and the users benefit greatly from ease of use. System that is easy to use lead to increase the users work satisfaction, and satisfied user increase the work productivity and actually increases business.
<table>
<thead>
<tr>
<th>Question 2</th>
<th>Does the highlighting of popular size helpful when making decision?</th>
</tr>
</thead>
</table>
| Result 2   | 70% – Totally agree  
|            | 30% - Partially agree  
|            | 0% - Neither Agree or Disagree  
|            | 0% - Partially Disagree  
|            | 0% - Totally disagree |

**Chart 2**

![Pie chart showing results](chart.png)

<p>| Analysis 2 | All the respondents agree with the question that the highlighting of the popular size is helpful during the decision making. 70% totally agree with the question and 30% partially agree with it. By having the highlighting of the size does help user to identify the popular size easily during the decision making. Besides that, it also managed to speed up the work done. Therefore we can conclude that is a good characteristic of GUI. |</p>
<table>
<thead>
<tr>
<th>Question 3</th>
<th>Does the filtering of popular size useful in decision making?</th>
</tr>
</thead>
</table>
| Result 3   | 60% – Totally Agree  
40% - Partially Agree  
10% - Neither Agree nor Disagree  
0% - Partially Disagree  
0% - Totally Disagree |
| Analysis 3 | 60% of the users are totally agree with the question that the filter features is helpful in the decision making, 30% of the users are partially agree with the question and another 10% neither agree nor disagree with it. Based on the analysis, we can conclude that with good feature, it will able to help users to accomplish their task effectively when using the system. |
**Question 4**  
The information provided is easy to understand

**Result 4**  
100% - Totally Agree  
0% - Partially Agree  
0% - Neither Agree nor Disagree  
0% - Partially Disagree  
0% - Totally Disagree

**Chart 4**

![Chart showing survey results]

**Analysis 4**  
The goal of Question 4 is to know whether the contents in the system are easy to understand. Majority of the users (70%) are totally agree with the idea and 30% of the users are partially agree with it. This can conclude that the information is presented in easy-to-understand format and in non-technical language.
<table>
<thead>
<tr>
<th>Question 5</th>
<th>I feel comfortable using this system</th>
</tr>
</thead>
</table>
| Result 5   | 50% – Totally Agree  
50% – Partially Agree  
0% - Neither Agree nor Disagree  
0% - Partially Disagree  
0% - Totally Disagree |

<table>
<thead>
<tr>
<th>Chart 5</th>
</tr>
</thead>
</table>

<p>| Analysis 5 | The goal of the question is to know whether users are comfortable using the system after applying HCI approach into the system. Based on the analysis done, it is found that all of them agree that the system are comfortable to work with, in which 50% are totally agree and another 50% are partially agree with the question. |</p>
<table>
<thead>
<tr>
<th>Question 6</th>
<th>The system gives error messages that clearly tell me how to fix problems</th>
</tr>
</thead>
</table>
| Result 6 | 60% – Totally Agree  
              20% - Partially Agree  
              20% - Neither Agree not Disagree  
              10% - Partially Disagree  
              0% - Totally Disagree |

Chart 6

![Chart showing the results of the question](chart.png)

Analysis 6 Does the error message prompted in the system assist users in fixing problem? This is the questions that the researcher wants to find out from the user. 60% of the users totally agree that the error message prompted help them to rectify the problem, 20% are partially agree with the questions, 20% are neither agree nor disagree and another 10% are partially disagree with it.
<table>
<thead>
<tr>
<th>Question 7</th>
<th>Does the solution provided by the system accurate?</th>
</tr>
</thead>
</table>
| Result 7   | 70% – Totally Agree  
30% - Partially Agree  
0% - Neither Agree nor Disagree  
0% - Partially Disagree  
0% - Totally Disagree |
<p>| Chart 7    | <img src="chart.png" alt="Pie Chart" /> |
| Analysis 7 | The goal of Question 7 is to know whether solutions given by the system are accurate. Majority of the users (70%) are totally agree with the idea and 30% of the users are partially agree with it. This can conclude that the solutions in CBR are good and shows the ability to learn. |</p>
<table>
<thead>
<tr>
<th>Question 8</th>
<th>Does the 5 features sufficient to make decision?</th>
</tr>
</thead>
</table>
| Result 8 | 40% – Totally Agree  
30% - Partially Agree  
20% - Neither Agree nor Disagree  
10% - Partially Disagree  
0% - Totally Disagree |
| Chart 8 | ![Pie Chart](chart.png) |
| Analysis 8 | Majority, which is 70% of the users agree that the 5 features are useful to make decision. 10% of the users does not agree with the question and said that they are not enough and should be revise. |
Question 9: Do you find the solution provided by the system useful?

Result 9:
- 50% – Totally Agree
- 40% - Partially Agree
- 0% - Neither Agree nor Disagree
- 10% - Partially Disagree
- 0% - Totally Disagree

Chart 9:

Analysis 9: 90% of the users agree that the solutions provided by the system are useful and 10% of them disagree with the idea. Therefore we can conclude that CBR is useful in explaining or justifying a solution.
<table>
<thead>
<tr>
<th>Question 10</th>
<th>Does the data collected help the company understand the customer better?</th>
</tr>
</thead>
</table>
| Result 10   | 70% – Totally Agree  
20% - Partially Agree  
10% - Neither Agree nor Disagree  
0% - Partially Disagree  
0% - Totally Disagree |
| Chart 10    | ![Pie Chart](image) |
| Analysis 10 | The main purpose of Question 10 is to find out whether the data in the system has the functions and capabilities that users had expected. 70% ranked totally agree as their choice, 20% opted partially agree as the choice and another 10% neither agree nor disagree with the question. This shows that the system developed had the features and capabilities which users had expected it to have. |
Question 11: Does the information provided in the system help the company to serve the customer better?

Result 11:
- 80% – Totally Agree
- 20% - Partially Agree
- 0% - Neither Agree nor Disagree
- 0% - Partially Disagree
- 0% - Totally Disagree

Chart 11

Analysis 11: All the users are satisfied with the system, in which 80% totally agree with the question and another 20% partially agree with it. This shows that the information in the system is useful and manages to help the company to provide a better service to the customer.

As at 24th January 2006, 14 test cases have been executed and 10 of them (71.4%) have been passed. There were 4 incidents found (3 medium and 1 low), of which all of them were already rectified by the researcher, retested and confirmed passed by the user.

<table>
<thead>
<tr>
<th>Table 8.15: Acceptance Testing Status Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Test Cases</td>
</tr>
<tr>
<td>14</td>
</tr>
</tbody>
</table>
Based on the user acceptance testing, the researcher found out that the system developed had meet the user requirements and expectations and be able to help user to accomplish their task efficiently and thus be able help the company realized its mission and goal.
8.5 CONCLUSION

Based on the testing that was done, it was found that all the system components work together as required by the system requirement. From the user acceptance testing, it was found that majority of the users are satisfied with the system but there are few improvements need to be applied to this system which later is discusses at Chapter 7. All of the testing conducted allows the researcher to detect errors and to ensure that the quality and reliability of program code from the earliest stages of development until the final stage and reduces the potential for errors.
CHAPTER 9

CONCLUSION

9.0 INTRODUCTION

The research and development work highlights the development of CRM software that was based on Human Computer Interaction (HCI) approach in order to address the CRM issue of usability use. This chapter will summarize the strengths of the system, system limitation and future works recommended for enhancement.

9.1 FULFILLMENT OF THE OBJECTIVES

In Chapter 1, two main objectives of the research project had been established:

Objective 1: To develop a usable Customer Relationship Management (CRM) system approach using User Centered Design (UCD) in order to increase the usability.

Fulfillment 1: The first objective of the research project was fulfilled where by CRM software had been developed using User-Centered Design (UCD) approach and Hierarchical Task Analysis technique in Human Computer Interaction (HCI) approach. It stresses the importance of usability in the system, in which the interfaces are designed as simple as possible with ease-of-use. The system had get users to be involved in the early
stage of development work where their requirement and concerns had been gathered and analyzed. Thus, it will be able to address the issue of user resistance. Therefore by combining those technique and approach provides a comfort level for the users that then lead to user satisfaction.

**Objective 2:** To incorporate Case Base Reasoning (CBR) in developing a Customer Relationship Management (CRM) system in decision making process.

**Fulfillment 2:** The second objective of this research had been fulfilled when Case Base Reasoning (CBR) using the nearest neighbor concept had been incorporated inside the decision making process in the CRM system. It is efficient to solve problem by starting with solution to a previous similar problem rather than generating a totally new solution again. Thus, it will be able to address the issue of system with ease-of-use.
9.2 FUTURE ENHANCEMENT

Based on the study and development experience encountered in this research, it is suggested that the CRM system can be scaled in terms of functionality. The following suggestions provide examples as to how the new system can be enhanced in this way.

1. Web-based application

A windows-based application is not enough for the CRM software. It should be enhanced to be made mobile by building the web-based application. The web-based application has many improvements in security therefore should be able to address the issue of unsecure data. The web-based CRM will be more manageable, easier for deployment, cost reduction and provide secure live data.

2. Product distribution module is not covered inside the system

The distribution of product to the customer was not covered in this scope of research project. By having the distribution of the product thus will be able to help customer to keep track of their order and thus will benefit both customer as well as the supplier.

3. View of graph

The viewing of graph is only on static mode. The current features only display the tyres based on the popularity and the zooming facility is not incorporate inside. It would be more interesting to have the live viewing so that users would be able to view the details of the tyre.
4. Graphical and comprehensive forecast reports

The graphical and comprehensive reports should be made online that is able to provide benefits the users. By having the comprehensive reports, it should improved forecast accuracy even if the input sales are limited.

9.3 CONCLUSION

This chapter wraps out the research project that had been carried out in the thesis. The CRM current issues had been identified and addressed accordingly by using HCI approach. Based on the research conducted, it managed to conclude that by using the User Centered Design (UCD) approach, Hierarchical Task Analysis (HTA) technique and other Human Computer Interaction (HCI) approach thus able to help user in performing their tasks in forecasting process, efficiently and effectively. An interactive and easy-to-use system is managed to be developed. The use of CBR which had been implemented successfully, manage to help the company to automate their decision making process. The objectives that had been identified in Chapter 1 – Introduction, had been managed to be achieved successfully.
REFERENCES


Adams, D. White paper: Data Visualization.


http://www.newtilt.com/about/pubs/user_adoption.pdf


[http://searchcrm.techtarget.com/originalContent/0,289142,sid11_gci940572,00.html](http://searchcrm.techtarget.com/originalContent/0,289142,sid11_gci940572,00.html)


January 10, 2006, from http://research.microsoft.com/~simonpj/papers/excel/matrix-
study.pdf#search=%22Champagne%20Prototyping%3A%20A%20Research%20Techniq
ue%20for%20Early%20Evaluation%20of%20Complex%20End-
User%20Programming%20Systems%22


http://research.ittoolbox.com/white-
1&p=511&h1=511&sp=

Guide. Addison-Wesley.

Completeness: A Key to Effective Net-based Customer Service System. Communications
of the ACM. 46(6), 47 - 51

from http://www.bsg.co.uk/solutions/business_need_user_experience.aspx

Introduction and Overview. Communications of the ACM.


Georgouspoulos, C. & Rana, O.F. (2002). An *approach to conforming a MAS into a FIFA-compliant system*. Communications of the ACM.


Hammond, K.J.(1986). Proceeding from American Association for Artificial Intelligence, AAAI-86, Philadelphia, PA, US. *CHEF: A Model of Case-Based Planning*


IPNet Solutions Inc (2002). E-Commerce Initiatives: The Supplier’s Perspective. IPNet


Morris, C.G., Maisto, A.A. Psychology: An Introduction. Prentise Hall.


Oracle Database 10g – Information is Currency. Is Your IT Investment Protected?


The deal with data. Retrieved February 10, 2007, from
http://www.themanufacturer.com/us/content/2718/The_deal_with_data

The smart organization’s guide to implementing change. (2005). Retrieved from August 9, 2007 from
http://www.hesketh.com/publications/the_smartorganizations_guide_to_implementing_change.html


University of Maryland. CHARMS-Choosing Human-Computer Interaction (HCI) Appropriate Research Methods.


http://www.teced.com/services_design_user_detail.html
http://www.softwarebusinessonline.com/eNewsletters/2007/newsletter_mar07.htm#us

What is Methodology?. Edward V. Berard. Retrieved December 1, 2005, from
http://www.toa.com/pub/methodology_article.txt

What is UML. Retrieved October 29, 2005, from
http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/what_is.uml.htm


